



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE CIENCIAS**

**ESCUELA DE CIENCIAS QUÍMICAS**

**INGENIERÍA EN BIOTECNOLOGÍA AMBIENTAL**

**" DESARROLLO DE SOFTWARE PARA EL DIMENSIONAMIENTO  
DE PLANTAS DE TRATAMIENTO DE AGUAS RESIDUALES "**

**TESIS DE GRADO**

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE:**

**INGENIERO EN BIOTECNOLOGÍA AMBIENTAL**

**AUTOR: CRISTIAN ANDRÉS SUÁREZ VILLALVA**

**TUTOR: DR. WILIAN BRAVO**

**RIOBAMBA - ECUADOR**

**2014**

## **AGRADECIMIENTOS**

A mi familia por su apoyo incalculable a lo largo de cada etapa de mi vida, especialmente, a mi madre que ha sido símbolo de fortaleza y lucha; y a mis abuelitos que han sido pilares fundamentales en mi desarrollo por su ejemplo y sus valores inculcados.

A mi novia que durante esta última etapa de mi vida fue una fuente de inspiración, alegría y perseverancia.

Al Dr. William Bravo y al Ing. Fernanda Rivera por la confianza, paciencia y ayuda brindada para el desarrollo del presente proyecto.

A los ingenieros Augusto Ferrari, Pablo Romero y Rafael Avalos por su colaboración brindándome sus conocimientos para el desarrollo del proyecto.

A mis compañeros y amigos que de una u otra forma siempre me apoyaron y confiaron en mí.

Y un agradecimiento especial para todos los docentes que más allá del deber de profesores supieron ser verdaderos maestros y amigos, dejando no solo su conocimiento sino también lecciones de vida.

**Cristian**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE CIENCIAS**

**ESCUELA DE CIENCIAS QUÍMICAS**

El Tribunal de Tesis certifica que: El trabajo de investigación:

**“DESARROLLO DE SOFTWARE PARA EL DIMENSIONAMIENTO DE PLANTAS DE TRATAMIENTO DE AGUAS RESIDUALES”**, de responsabilidad del señor egresado Cristian Andrés Suárez Villalva, ha sido prolijamente revisado por los Miembros del Tribunal de Tesis, quedando autorizada su presentación.

FIRMA

FECHA

Ing. César Ávalos

**DECANO DE LA  
FACULTAD DE CIENCIAS**

\_\_\_\_\_

\_\_\_\_\_

Dra. Nancy Veloz

**DIRECTORA ESC.  
CIENCIAS QUÍMICAS**

\_\_\_\_\_

\_\_\_\_\_

Dr. Wilian Bravo

**DIRECTOR DE TESIS**

\_\_\_\_\_

\_\_\_\_\_

Ing. Fernanda Rivera

**MIEMBRO DE TRIBUNAL**

\_\_\_\_\_

\_\_\_\_\_

**DIRECTOR CENTRO DE  
DOCUMENTACIÓN**

\_\_\_\_\_

\_\_\_\_\_

**NOTA DE TESIS**

\_\_\_\_\_

Yo, **Cristian Andrés Suárez Villalva**, soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis; y el patrimonio intelectual de la Tesis de Grado, pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

---

**Cristian Andrés Suárez Villalva**

# TABLA DE CONTENIDO

## RESUMEN

## SUMMARY

## INTRODUCCIÓN

## OBJETIVOS

<b>CAPITULO I</b> .....	<b>- 1 -</b>
<b>1. MARCO TEÓRICO</b> .....	<b>- 1 -</b>
1.1. Aguas Residuales.....	- 1 -
1.1.1. Clasificación de Aguas Residuales .....	- 2 -
1.1.2. Características de las Aguas Residuales.....	- 2 -
1.2. Tratamiento de Aguas Residuales .....	- 8 -
1.2.1. Tratamiento Primario .....	- 9 -
1.2.2. Tratamiento Secundario.....	- 9 -
1.2.3. Tratamiento Terciario.....	- 10 -
1.3. Software .....	- 11 -
1.3.1. Clasificación del Software.....	- 12 -
1.3.2. Proceso de Creación de Software .....	- 13 -
1.3.3. Fases del Desarrollo del Software .....	- 19 -
1.3.4. Licencia de Software .....	- 21 -
<b>CAPITULO II</b> .....	<b>- 27 -</b>
<b>2. MATERIALES Y MÉTODOS</b> .....	<b>- 27 -</b>
2.1. Materiales Y Equipos.....	- 27 -
2.2. Métodos.....	- 27 -
2.2.1. Método de Selección de Procesos.....	- 27 -
2.2.2. Método para el Diseño de dos Procesos .....	- 28 -

2.2.3.	Metodología para la Programación .....	- 28 -
2.2.4.	Metodología de Validación de la Herramienta.....	- 28 -
CAPITULO III .....		- 30 -
3.	DESARROLLO Y RESULTADOS.....	- 30 -
3.1.	Desarrollo de los Procesos de Tratamiento .....	- 30 -
3.1.1.	Desarrollo del Canal Parshall .....	- 30 -
3.1.2.	Desarrollo de las Rejillas .....	- 31 -
3.1.3.	Desarrollo del Desarenador Horizontal .....	- 33 -
3.1.4.	Desarrollo del Filtro Anaerobio de Flujo Ascendente "FAFA" .....	- 36 -
3.1.5.	Desarrollo del Tanque Imhoff.....	- 38 -
3.1.6.	Desarrollo de la Laguna Anaerobia.....	- 40 -
3.1.7.	Desarrollo de la Laguna Aerobia.....	- 42 -
3.1.8.	Desarrollo de la Laguna Facultativa.....	- 42 -
3.2.	Desarrollo del Software .....	- 44 -
3.2.1.	Desarrollo de la Programación Interna.....	- 44 -
3.2.2.	Desarrollo de la Interface de Cada Proceso.....	- 47 -
3.2.3.	Desarrollo de la Interface General .....	- 52 -
3.2.4.	Desarrollo del Instalador .....	- 54 -
3.3.	Resultados.....	- 55 -
<b>CONCLUSIONES.....</b>		<b>- 62 -</b>
<b>RECOMENDACIONES.....</b>		<b>- 63 -</b>
<b>BIBLIOGRAFÍA</b>		
<b>ANEXOS</b>		

## ÍNDICE DE FIGURAS

<b>Ilustración 1:</b> Vista Planta del Canal Parshall .....	- 30 -
<b>Ilustración 2:</b> Vista Lateral del Canal Parshal .....	- 31 -
<b>Ilustración 3:</b> Modelo ilustrativo de una Rejilla.....	- 32 -
<b>Ilustración 4:</b> Vista frontal de un Desarenador Horizontal.....	- 34 -
<b>Ilustración 5:</b> Vista lateral de un Tanque FAFA .....	- 37 -
<b>Ilustración 6:</b> Vista frontal de un Tanque Imhoff .....	- 39 -
<b>Ilustración 7:</b> Vista lateral de una zanja para lagunaje .....	- 41 -
<b>Ilustración 8:</b> Ejemplo de la definición de variables dentro de la programación .....	- 46 -
<b>Ilustración 9:</b> Ejemplo de la programación para que el programa realice los cálculos-	46 -
<b>Ilustración 10:</b> Ventana de WindowsForms en blanco.....	- 48 -
<b>Ilustración 11:</b> Ventana de WindowsForms con la interface ya realizada .....	- 48 -
<b>Ilustración 12:</b> ToolTip usado para proporcionar información .....	- 49 -
<b>Ilustración 13:</b> Ejemplo de la programación usada para leer los datos ingresados por el usuario.....	- 49 -
<b>Ilustración 14:</b> Línea de código usada para definir el formato en que se realizara el guardado .....	- 50 -
<b>Ilustración 15:</b> Ejemplo de la programación realizada par amostrar los resultados de los cálculos.....	- 50 -
<b>Ilustración 16:</b> Programación usada para definir condiciones dentro de los cálculos y los mensajes de error .....	- 51 -
<b>Ilustración 17:</b> Menú principal del software PTAR .....	- 52 -
<b>Ilustración 18:</b> Ejemplo de la programación usada para agregar las descripciones...	- 53 -
<b>Ilustración 19:</b> Programación de botones para que ejecuten el proceso correspondiente -	53 -
<b>Ilustración 20:</b> Diagrama de flujo del proceso de creación del Software .....	- 55 -
<b>Ilustración 21:</b> Grafico que muestra el grado de satisfacción con la facilidad de uso del software .....	- 56 -

<b>Ilustración 22:</b> Grafico que muestra el grado de satisfacción con el uso de la interface del software .....	- 57 -
<b>Ilustración 23:</b> Grafico que muestra el grado de satisfacción con la estética de la interface del software.....	- 57 -
<b>Ilustración 24:</b> Grafico que muestra el grado de satisfacción del uso del programa PTAR-	57 -
<b>Ilustración 25:</b> Grafico que muestra la satisfacción con la manera en que el programa presenta los resultados.....	- 58 -
<b>Ilustración 26:</b> Grafico del grado de satisfacción con la exactitud de los resultados..	- 58 -
<b>Ilustración 27:</b> Grafica del grado de satisfacción con las características del software-	59 -
<b>Ilustración 28:</b> Grafico del grado de satisfacción con el método de guardado de los cálculos.....	- 59 -
<b>Ilustración 29:</b> Grafico de la satisfacción con el formato de impresión de los resultados del software .....	- 60 -

## RESUMEN

Se desarrolló un software para el dimensionamiento de plantas de tratamiento de aguas residuales, en la Escuela de Ciencias Químicas perteneciente a la Facultad de Ciencias de la Escuela Superior Politécnica de Chimborazo; con el objetivo de facilitar y agilizar los diferentes cálculos para dimensionar los procesos de una planta de tratamiento.

Se estudió la problemática presente en los laboratorios de la facultad los que requieren de este tipo de herramientas para un mejor desarrollo de las actividades y del aprendizaje, se hizo un análisis para determinar que procesos de tratamiento deben ser incluidos en el programa; también se determinó la mejor metodología de desarrollo de software para este proyecto, se usó la metodología Ágil XP aplicando el paradigma de programación orientado a objetos; finalmente se analizaron los métodos de comprobación y validación de la herramienta llegando a utilizar el Método Delphi conjuntamente con la Escala de Likert para la obtención de resultados tanto cualitativos como cuantitativos.

Se obtuvo así un software de dimensionamiento de plantas de tratamiento de aguas residuales de uso didáctico que presenta características de Fácil Uso 92%, Interface Amigable 87%, Interface Agradable 75%, Comprensión del Programa 96%, Comprensión de los Resultados que proporciona el programa 92%, Exactitud de los Resultados del Programa 75%, Características del Programa 50%, Guardado de Resultados 58%, Formato para Imprimir 67%.

Se recomienda su implementación en los laboratorios de la Escuela de Ciencias Químicas, una difusión entre los profesores de la facultad; también se recomienda poner a disposición el programa para todas las instituciones y empresas que desarrollen actividades relacionadas con plantas de tratamiento de aguas residuales.

## **SUMMARY**

Was developed a Software for sizing of wastewater treatment plants, in the School of Chemical Sciences belonging to the Faculty of Sciences in Escuela Superior Politécnica de Chimborazo; with the aim to facilitate and expedite the different calculations to size processes of treatment plant.

Was studied the problems present in the laboratories of the faculty who require this type of tools for better development of activities and learning, an analysis was made to determine what treatment processes should be included in the program; the best software development methodology for this project was also determined, was used Agile XP methodology applying the object-oriented programming paradigm; finally was analyzed the methods of testing and validation of the tool getting to use the Delphi Method together with Likert Scale to obtain both qualitative and quantitative results.

Was obtained a didactic use Software for sizing of wastewater treatment plants having features of Easy Use 92%, Friendly Interface 87%, Pleasant Interface 75%, Program Comprehension 96%, Understanding the results provided by the program 92%, Program Results Accuracy 75%, Program Features 50%, Results Saving 58%, Printable Format 67%.

Is recommended its implementation in the laboratories of the School of Chemical Sciences, a diffusion between faculty members; is also recommended make available the program for all institutions and companies who develop plants wastewater treatment related activities.

## **INTRODUCCIÓN**

### **Situación Problemática y Justificación**

En los cuerpos de agua naturales la calidad de su agua se ve afectada por varios factores especialmente humanos entre ellos la descarga de aguas residuales urbanas; estos sistemas eran capaces de depurar los contaminantes de manera natural, pero con el incremento de la población y la producción industrial, se elevan también el caudal y la carga contaminante que reciben estas aguas; lo que ha llevado a que los cuerpos de agua naturales en los que se vierten las aguas urbanas, pierdan esa capacidad auto depuradora.

La eliminación de aguas residuales tanto domésticas como industriales en los distintos sistemas de aguas se traduce en un problema ambiental e inclusive un problema de salud pública; debido a que en la mayoría de casos éstas aguas incluye en su contenido una variedad muy grande de compuestos orgánicos, inorgánicos y microorganismos muchas veces patógenos que modifican la calidad del agua natural y a la vez que vuelven a este valioso recurso en una clara fuente de enfermedades para la gente, animales y plantas.

Para el diseño de una planta de tratamiento de aguas residuales domésticas e industriales se debe analizar una variedad de factores que pueden afectar el dimensionamiento, desempeño y la confiabilidad de los procesos de dimensionamiento diseñados; es indispensable para ello determinar caudales y la carga contaminante de diseño, para de esta manera poder escoger las unidades de tratamiento apropiadas; además siempre teniendo en cuenta factores ambientales, económicos y sociales; y de esta manera lograr un efluente que logre unos parámetros bajo la normativa ambiental.

Por lo tanto el tratamiento de las aguas residuales constituye una gran alternativa de solución para mitigar los daños producidos por la contaminación en los cuerpos de agua receptores, lo que ayuda de gran manera al cuidado de la salud pública y a la protección del ambiente. Sin embargo, ningún programa de control

tendría éxito si no se cuenta con los recursos financieros, técnicos, intelectuales y tecnológicos; para su diseño, implementación, operación y mantenimiento.

Aquí es donde la tecnología toma su importancia en cada aspecto personal y laboral, constituyendo herramientas que facilitan y agilitan nuestro día a día. Si a esto le sumamos el desarrollo acelerado de la tecnología con lo cual se vuelve casi imprescindible el uso de las mismas para mantenernos al nivel competitivo global, se vuelve inminente e indispensable el desarrollo de estas tecnologías acorde a nuestras propias necesidades.

Entre tan variados tipos de tecnologías una de gran importancia y que es básicamente la que da vida a todas las demás es el "Software", que son conjuntos de instrucciones lógicas que nos permiten interactuar con equipos y así mismo nos ayudan a organizar y procesar datos, este procesamiento de datos puede ser utilizado ayudándonos en el desarrollo y diseño de diferentes sistemas que vemos en el campo laboral por lo cual constituye la principal herramienta de trabajo para quienes trabajamos con datos como la base para la toma de decisiones.

## **OBJETIVOS**

### **General**

- Desarrollar un software que automatice el dimensionamiento de plantas de tratamiento de aguas residuales.

### **Específicos**

- Diseñar cada una de las diferentes etapas del tratamiento de aguas residuales de manera adecuada.
- Gestionar el diseño de una planta de tratamiento de aguas en una interface intuitiva para un fácil manejo.
- Proveer un sistema que se ajuste a las necesidades o características de diseño requeridas por cada profesional.

# CAPITULO I

## 1. MARCO TEÓRICO

### 1.1. Aguas Residuales

El término agua residual es usado para referirse a un tipo de agua que se encuentra con contaminación por la presencia de sustancias de desecho ya sean estas domésticas o industriales. Su relevancia y peligrosidad es tanta que se requiere de sistemas de canalización y tratamiento para ser eliminadas. La falta de tratamientos o tratamientos indebidos conlleva severos problemas tanto de contaminación como de salud.

A las aguas residuales también suelen ser llamadas aguas servidas, aguas grises, aguas fecales o aguas cloacales. Son residuales porque después de haber sido usada el agua pasa a constituir un residuo, que bajo ninguna circunstancia debe ser usada de modo directo; y son transportadas mediante cloacas o lo comúnmente llamado alcantarilla en nuestro medio.

Varios autores diferencian las aguas servidas de las aguas residuales ya que las primeras serían solo aquellas que se genera del uso doméstico y las segundas vienen a ser la mezcla de todo tipo de aguas incluidas las domésticas y las industriales. Sin importar como se las denomine, generalmente en todos los casos, están conformadas por todo tipo de aguas que se encuentran y transportadas en el alcantarillado público y que incluyen, generalmente, las aguas provenientes de la lluvia y las infiltraciones en el terreno de todo tipo de aguas.

La composición y su tratamiento pueden variar mucho de un caso a otro, por lo cual en los residuos industriales se sugiere la depuración en el origen del vertido que un tratamiento conjunto posterior.

### **1.1.1. Clasificación de Aguas Residuales**

En función del origen de las aguas, puede clasificarse a las aguas residuales en:

- a) **Domésticas:** son básicamente residuos humanos en las que el agua ha sido utilizada con fines higiénicos (baños, cocinas, lavanderías, etc.), y que llegan a las redes de alcantarillado público a través de las descargas de instalaciones hidráulicas de los hogares y así como las generadas en lugares de tipo comercial o públicos.
- b) **Industriales:** son líquidos generados en los procesos industriales. Mismos que por el uso especial que se le da posee características específicas, según cada tipo de industria.
- c) **Infiltración y caudal adicionales:** las aguas de infiltración son aguas que generalmente se calan en el sistema de alcantarillado ya sea por debilidad o defectos en la estructura de las tuberías, así como también durante la limpieza o inspección de los sistemas de tuberías, etc.

Los caudales adicionales son los que provienen de la lluvia y que son evacuados con ayuda de los sistemas urbanos como canales, drenajes y colectores.

- d) **Pluviales:** lo que se conoce comúnmente como agua de lluvia, la lluvia es responsable de descargar grandes cantidades de agua al suelo. En el casco urbano gran parte de esta agua es drenada uniéndose a las aguas domésticas en el sistema de alcantarillado y otra parte escurre por la superficie, arrastrando consigo todo tipo de residuos y tierra que se pueden encontrar sobre el suelo.

### **1.1.2. Características de las Aguas Residuales**

Los elementos que se encuentran en las aguas residuales suelen ser clasificados como: físicos, químicos y biológicos; la caracterización de las aguas residuales es de suma importancia para poder establecer el grado de contaminación y principalmente las cargas orgánicas y de sólidos que transportan, además de que con ello se puede diagnosticar la contaminación provocada por el vertimiento en los cuerpos de agua y determinar las operaciones y procesos de tratamiento que proporcionen los mejores resultados y que ofrezcan un mayor ahorro en cuanto a costos.

### 1.1.2.1. Características Físicas.

Las características físicas más importantes en la caracterización de un agua residual son: los sólidos totales, la turbiedad, el color, el olor, la temperatura y la conductividad.

- **Sólidos:** Se clasifica toda la materia, excepto el agua contenida en los materiales líquidos, como materia sólida.

Precisar los niveles de sólidos tanto de sólidos volátiles como de sólidos totales es importante para estimar la presencia de estos en el agua residual y para limitar la eficiencia que puedan presentar las unidades de procesamiento del agua; la estimación de los sólidos sedimentables es la base para establecer la necesidad de procesos de sedimentación, tanto como tratamiento como para asegurar la eficiencia de los demás procesos.

- **Turbidez:** Es una propiedad de tipo óptico en la que una suspensión no permite que la luz se transmita a través del fluido; este fenómeno óptico puede estar ligado a una gran variedad de materiales que se encuentran suspendidos en el líquido y que pueden variar tamaño, desde dispersiones coloidales hasta partículas gruesas, entre las que podemos encontrar tenemos tierra, arena, materia orgánica e inorgánica muy pequeña y todo tipo de microorganismos.

- **Color:** El color que presentan las aguas residuales se debe a la presencia de todo tipo de sustancias desde sólidos suspendidos hasta sustancias en solución. El color provocado por la presencia sólidos en suspensión se llama color aparente, en tanto que el color verdadero es causado por sustancias disueltas y coloidales. Cualitativamente hablando, el color se puede usar para describir de manera general la condición en la que se encuentra el agua residual aunque esta determinación es solo una estimación. Si el agua presenta un color café claro, se puede concluir que el agua lleva aproximadamente unas 6 horas desde su descarga. Si el agua se observa de un color gris claro significa que el agua ha sufrido por algún tipo de degradación o que han estado por algún tiempo en los sistemas de alcantarillado. Si por el contrario las aguas presentan un color gris oscuro o negro, significa generalmente que el agua han sufrido un alto grado de descomposición provocado por las bacterias bajo condiciones anaerobias. El oscurecimiento de las aguas residuales se debe generalmente a la presencia de sulfuros, en especial el sulfuro ferroso (FeS).

- **Olor:** Determinar el olor es una característica cada vez más valiosa, primero porque ayuda a identificar condiciones específicas del agua como la presencia de ciertos elementos en la misma y segundo porque constituye una característica importante para la población ya que el olor es el primer elemento que puede molestar a las personas más que las operaciones que se realizan en las instalaciones de una planta de tratamiento de aguas residuales. El agua residual recién liberada generalmente no presenta un olor agresivo o peligroso, pero con el paso del tiempo y por efectos de la degradación biológica que se da bajo condiciones anaeróbicas, se liberan una gran variedad de compuestos que presentan mal olor. El compuesto más común con esta característica es el sulfuro de hidrógeno.

- **Temperatura:** La adición de aguas domésticas y especialmente aguas industriales provoca que las aguas residuales tengan una mayor temperatura que el agua de uso urbano, ya que por lo general al momento de ser descargadas estas presentan temperaturas altas especialmente las industriales por los usos que se le puede dar al agua antes de ser eliminada. Las reacciones químicas así como la velocidad de reacción son fuertemente afectadas por la temperatura por ello es un parámetro muy importante en el agua, además de que la vida acuática es fuertemente afectada por la temperatura; también es un parámetro importante para la agricultura ya que la temperatura del agua de riego puede afectar a las plantas.

- **Conductividad:** La conductividad del agua es una característica cuantitativa de su capacidad para transferir una corriente eléctrica a través de su masa, la misma que depende de la cantidad total de sustancias ionizadas presentes en el agua así como de la temperatura en la cual se realice la medición. Por lo cual, todo cambio en la cantidad de las sustancias presentes, en la movilidad o valencia de sus iones disueltos, significa un cambio en el valor de conductividad, por esta razón la determinación de la conductividad se suele realizar in situ y en el análisis del agua este valor suele dar un estimado rápido del contenido de sólidos.

### 1.1.2.2. Características Químicas.

A los constituyentes químicos de las aguas residuales con frecuencia los podemos clasificar en inorgánicos y orgánicos. Los inorgánicos incluyen todo tipo de elementos individuales como: calcio, hierro y una gran gama de compuestos como nitratos y sulfatos. Los compuestos orgánicos presentes de mayor relevancia se pueden clasificar en agregados e individuales, los agregados están conformados por un número de compuestos a los que no puede detallar de manera individual.

- **Materia inorgánica:** Las aguas residuales y naturales presentan una amplia variedad de elementos inorgánicos especialmente aquellos que son importantes para la determinación y control de la calidad del agua. Las concentraciones de las sustancias inorgánicas en el agua aumentan tanto por el contacto del agua con las diferentes formaciones geológicas, como por las aguas residuales, tratadas o sin tratar, que a ella se descargan.

La concentración de varios elementos inorgánicos presentes en el agua puede afectar en gran medida a los usos que se le pueda dar al agua por esta razón es conveniente verificar la naturaleza de los más importantes, en especial de los que han sido agregados al agua durante su utilización.

- **pH:** La presencia del ion hidrógeno así, en especial la concentración del mismo constituyen un parámetro de mucha relevancia dentro de la calidad del agua ya sea para las aguas naturales así como para aguas residuales. El rango de concentración del ion hidrógeno adecuado para la correcta proliferación y el desarrollo adecuado de la vida es muy corto y bastante importante. Un agua residual con valores inadecuados de pH puede presentar dificultades de tratamiento bajo procesos biológicos, y las descargas pueden alterar el valor del pH en los cuerpos de agua receptores si ésta no se modifica antes de su descarga.

- **Cloruros:** El agua natural generalmente contiene cloruros que provienen del contacto con suelo y rocas que los contienen. La fuente más significativa de cloruros son las descargas de aguas residuales, por ejemplo, una persona puede aportar con alrededor de 6g de cloruros por día únicamente con sus heces. Debido a que los procesos de tratamiento de aguas comúnmente usados no abarcan la supresión de cloruros en cantidades importantes, determinar

cantidades de cloruros sobre los valores normales pueden significar un indicativo de que esa agua está siendo utilizada para la descarga de aguas residuales.

- **Alcalinidad:** La alcalinidad de un agua residual se debe a la presencia de hidróxidos, carbonatos y bicarbonatos de elementos como el calcio, el magnesio, el sodio, el potasio o el amoníaco. De entre todos ellos, los más comunes son el bicarbonato de calcio y el bicarbonato de magnesio. La alcalinidad del agua contrarresta los cambios de pH que pueden ser producidos por la adición de ácidos al agua. Generalmente las aguas residuales son alcalinas por los elementos agregados en sus diversos usos. La alcalinidad se expresa como un valor de carbonato de calcio,  $\text{CaCO}_3$ .

El valor de alcalinidad en un agua de tratamiento es importante cuando se va a tratar mediante químicos, cuando se van a eliminar nutrientes mediante procesos biológicos, y cuando se busca suprimir el amoníaco por arrastre con aire.

- **Nitrógeno:** Los elementos nitrógeno y fósforo son nutrientes esenciales para el crecimiento de protistas y plantas, pequeñas cantidades de otros elementos son necesarios para el crecimiento biológico; sin embargo, el nitrógeno y el fósforo constituyen para la mayoría de estos organismos su principal nutriente. Dado que el nitrógeno es un componente indispensable en la síntesis de proteínas, es muy importante obtener datos que indiquen si este está o no presente en el agua, y en qué valores, para estudiar y determinar las posibilidades de un tratamiento biológico para estas aguas. En el caso de que la cantidad de nitrógeno presente en el agua no sea la requerida para los procesos biológicos, será necesario añadir esos nutrientes. Por otro lado, puede ser necesario la eliminación del nitrógeno en aquellos caso que se requiera controlar el crecimiento de las algas presentes en el agua receptora y de esta manera evitar que se alteren sus características.

- **Fósforo:** En la actualidad es una preocupación el limitar la cantidad de compuestos de fosforo que llegan a los sistemas de agua naturales por medio de las descargas residuales, debido a que el fosforo también es un elemento esencial para el crecimiento de las algas, así como también de muchos otros organismos, y la presencia de este elemento podría degenerar en proliferaciones incontroladas de algas.

- **Azufre:** El Azufre o ion sulfato es un elemento que de manera natural se puede encontrar presente tanto en aguas naturales como en aguas residuales. Es

liberado con la degradación de las proteínas, las mismas que necesitan su presencia para poder ser sintetizadas. Los sulfatos se reducen químicamente a sulfuros y a sulfuros de hidrógeno (H<sub>2</sub>S) bajo la acción bacteriana en condiciones anaerobias.

- **Gases:** Los gases que se encuentran con mayor frecuencia en aguas residuales son el nitrógeno (N<sub>2</sub>), el oxígeno (O<sub>2</sub>) que suele encontrarse en bajas cantidades por lo cual generalmente se realizan procesos de aireación para promover procesos de degradación aerobia, el dióxido de carbono (CO<sub>2</sub>), estos tres son elementos comunes de la atmosfera y se encuentran presentes en el agua por estar en contacto con ella; el sulfuro de hidrógeno (H<sub>2</sub>S), el amoníaco (NH<sub>3</sub>), y el metano (CH<sub>4</sub>) también presentes en el agua se generan en los procesos de descomposición de la materia orgánica que se encuentra en las aguas residuales.

#### *1.1.2.3. Características Biológicas.*

Las características biológicas son muy importantes para el control de enfermedades causadas por microorganismos patógenos que pueden estar presentes en el agua y por lo valiosos que son ciertos microorganismos para los procesos biológicos del tratamiento de las aguas residuales.

Los principales grupos de microorganismos presentes en las aguas residuales así como superficiales se clasifican en organismos eucariotas, bacterias y arqueos bacterias.

#### - **Organismos Patógenos**

En las aguas residuales podemos encontrar gran variedad de organismos patógenos que provienen generalmente de desechos de humanos que se encuentran infectados. Las principales clases de organismos patógenos presentes en las aguas residuales son: bacterias, virus y protozoarios. Los microorganismos patógenos excretados por el hombre normalmente causan enfermedades del aparato intestinal como la fiebre tifoidea, paratifoidea, la disentería, diarrea y cólera. Estos organismos cada año son responsables de un

gran número de muertes en países con deficiencias sanitarias, las zonas tropicales especialmente como zonas de mayor incidencia.

#### - **Organismos Indicadores**

Los organismos patógenos se presentan en las aguas residuales contaminadas en cantidades variadas aunque generalmente su presencia es baja, y además, difícilmente se los puede aislar e identificar. Por ello se emplea el organismo coliforme como indicador, debido a que su presencia es más numerosa y fácil de comprobar. El tracto intestinal humano posee un sinnúmero de bacterias conocidas como organismos coliformes, cada humano elimina, junto con sus heces, un valor de entre 100 000 a 400 000 millones de organismos coliformes cada día. Es por esto que la presencia de organismos coliformes se puede considerar como un indicador de la probable existencia de organismos patógenos, y que la ausencia de los mismos es un indicador de que las aguas se encuentran libres de organismos peligrosos que pueden causar enfermedades. Aunque no se puede considerar un indicador de confianza absoluta ya que algunos patógenos pueden estar presentes en el agua aún si no se encuentran presentes los coliformes.

### **1.2. Tratamiento de Aguas Residuales**

El tratamiento de las aguas residuales, es un conjunto de procesos necesarios para evitar la contaminación física, química, bioquímica y biológica de los cuerpos y sistemas de agua receptores. De manera general, el tratamiento de aguas residuales busca evitar:

- La contaminación a los suministros de agua públicos, privados e industriales.
- La contaminación de las aguas designadas a la recreación y el esparcimiento.
- La contaminación de las aguas destinadas a las actividades piscícolas.
- Deterioro a la agricultura y pérdida del valor de la tierra.
- Daño al entorno ecológico.

Incluye una serie de operaciones unitarias que son los diferentes métodos de tratamiento en los que intervienen los fenómenos físicos, y procesos unitarios que

son métodos mediante los cuales se realiza una eliminación de los contaminantes usando como base procesos químicos o biológicos. En la actualidad, tanto las operaciones como los procesos unitarios se congregan para formar los así llamados tratamientos primarios, secundarios y terciarios.

### **1.2.1. Tratamiento Primario**

Tienen como objetivo principal el de eliminar o reducir los sólidos presentes en suspensión mediante los procesos de sedimentación ya sea por simple por gravedad o mediante una sedimentación asistida por coagulantes y floculantes.

Las estructuras que se encargan de esta función son los llamados estanques de sedimentación primarios o también conocidos como clarificadores primarios. Comúnmente se los diseña para eliminar las partículas que poseen tasas de sedimentación de 0,3 a 0,7 mm/s.

En esta etapa se consigue, por medio de la precipitación, eliminar un porcentaje entre el 60 y el 70% de los sólidos en suspensión. Es decir, el tratamiento primario es capaz de remover fracción importante de la carga orgánica ya que una gran cantidad de los sólidos suspendidos están conformados por materia orgánica aparte de tierras y arenas; y que puede llegar a significar entre el 25% y el 40% de la DBO y entre el 50% y el 65% de los sólidos suspendidos. Entre los tipos de tratamiento primario se citan:

- Sedimentación primaria.
- Flotación.
- Precipitación química.
- Filtros gruesos.
- Oxidación química.
- Coagulación, floculación, sedimentación y filtración.

### **1.2.2. Tratamiento Secundario**

Tiene como objetivo principal eliminar la materia orgánica ya sea que se encuentre en disolución o en un estado coloidal, usando para ello procesos de

oxidación biológica. Estos procesos biológicos son procesos naturales controlados en los que intervienen los organismos presentes en el agua residual, y que se desarrollan en una estructura que por su función se denomina reactor. Estos microorganismos, principalmente bacterias, se alimentan de la materia orgánica presente en el agua liberando anhídrido carbónico y agua, lo cual genera una biomasa bacteriana la misma que se sedimenta. Así, el agua queda libre de materia orgánica produciendo unos fangos para los que hay que diseñar un tratamiento apto para poder ser dispuestos.

El sedimento producto del tratamiento, como se dijo, está formado fundamentalmente por materia orgánica, partículas sedimentables y por bacterias, se denomina fango activo.

Los microorganismos del proceso de tratamiento se pueden encontrar en el agua suspendidos (procesos de crecimiento suspendido o fangos activados), ligados a un sustrato de suspensión (procesos de crecimiento adherido) o colocados en un sistema mixto (procesos de crecimiento mixto).

Las estructuras que comúnmente se usan para el tratamiento secundario incluyen entre otros: filtros de arena intermitentes, filtros percoladores, contactores biológicos rotatorios, lechos fluidizados, estanques de fangos activos, lagunas de estabilización u oxidación y sistemas de digestión de fangos.

### **1.2.3. Tratamiento Terciario**

El objetivo principal de estos procesos es suplementar los procesos previamente realizados para obtener descargas más puras, que sea menos contaminante y que a los que se les puedan dar variados usos como recarga de acuíferos, recreación, agua industrial, etc. Las sustancias o compuestos comúnmente removidos son: fosfatos y nitratos, huevos y quistes de parásitos, sustancias tensoactivas, algas, bacterias y virus (desinfección), sólidos totales y disueltos, temperatura.

### **1.3. Software**

Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático, en el que se integran grupos de elementos lógicos indispensables que lograr la realización de tareas determinadas.

Los componentes lógicos incluyen, entre varios más, las aplicaciones informáticas; tales como procesadores de texto, que facilitan al usuario para llevar a cabo todo tipo de tareas relacionadas a la edición de textos; el llamado software de sistema, conocido como el sistema operativo, que es la base fundamental que faculta al resto de los programas para que funcionen debidamente, favoreciendo de esta manera también el funcionamiento conjunto entre los componentes físicos y el resto de las aplicaciones, y de esta manera proporcionar una interfaz para el usuario.

Software es una palabra proveniente del inglés que en español no posee una traducción adecuada al contexto, por lo cual se la utiliza sin ningún tipo de traducción y así fue registrada por la Real Academia Española. Aunque puede no ser estrictamente lo mismo, suele reemplazarse con palabras afines tales como programas (informáticos) o aplicaciones (informáticas) o soportes lógicos.

Existen muchas descripciones para referirse a lo que es un software, pero posiblemente la más precisa y más adecuada sea:

Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación.

En base a esta descripción, la definición de software va por encima de los programas de computación en sus variados tipos: código fuente, binario o ejecutable; también su documentación, los datos a procesar e incluso la información de usuario forman parte del software: es decir, engloba todo aquello que no se puede tocar, todo lo que no posee un estado físico propio.

La palabra software fue usada por primera vez con este significado por John W. Tukey en 1957. En la ingeniería de software y las ciencias de la computación, en la que se describe al software como toda la información procesada por los sistemas informáticos: programas y datos.

La concepción de leer variadas series de instrucciones (programa) desde la memoria de un dispositivo para controlar los cálculos fue introducido por Charles Babbage como parte de su máquina diferencial. La teoría que forma la base de la mayor parte del software moderno fue propuesta por Alan Turing en su ensayo de 1936, Los números computables, con una aplicación al problema de decisión.

### **1.3.1. Clasificación del Software**

Orientado a lo práctico se ha clasificado al software bajo tres grandes conceptos:

- **Software de sistema:** El software de sistema le provee al usuario y programadores de interfaces apropiadas que estén a la altura del usuario, controladores, herramientas y utilidades de apoyo que faciliten el cuidado del sistema global.

Por ejemplo bajo este concepto se encuentran los sistemas operativos y los controladores o drivers de dispositivos y accesorios.

- **Software de programación:** Son programas desarrollados y orientados a facilitar el desarrollo de otros programas, teniendo como base varias alternativas y conjuntos completos de lenguajes de programación, que permitan el desarrollo de una manera práctica. Engloban fundamentalmente Editores de texto, Compiladores, Intérpretes, Enlazadores, Depuradores y Entornos de Desarrollo Integrados.

- **Software de aplicación:** Son aquellos programas que facultan al usuario para realizar una o varias tareas determinadas, en todo tipo de actividades y campos que puedan ser automatizado o asistido.

Incluye entre muchos otros: Software educativo, Software empresarial, Videojuegos, Software de cálculo numérico y simbólico, Software de diseño asistido (CAD).

### **1.3.2. Proceso de Creación de Software**

#### **1.3.2.1. Captura, Análisis Y Especificación De Requisitos.**

De manera fácil y sencilla esta fase incluye todo lo que es obtención de requisitos y características que se desea o se espera para el programa ya sean estas funcionales o no funcionales, las mismas que al final del desarrollo se espera que cumpla el programa.

Las distintas características, que poseerá el sistema o programa a desarrollar, así como de su interface y entorno, parámetros no funcionales y arquitectura, y sobre todo los beneficios de estas características; dependen en su mayoría de que tan bien se desarrolle esta etapa. Esta es, muy probablemente, la etapa de mayor importancia y a la vez una de las más complicadas de llevar a cabo satisfactoriamente, pues no es automatizable, no posee una técnica bien definida y depende mucho de la habilidad y experticia del profesional que la realice.

Involucra directamente al usuario o cliente del sistema, si bien se han ideado varias metodologías, incluso existe software de apoyo, para la captura, elicitación y registro de requisitos, no existe una forma eficaz o completamente confiable para llevar esta etapa, y deben ponerse en práctica criterios fiables y sobre todo sentido común por parte de quien o quienes realicen esta etapa; también es de vital importancia generar una clara y apropiada relación y comprensión con el cliente o usuario final.

El resultado más importante de la finalización de esta fase es lo que se denomina generalmente como especificación de requisitos software o simplemente el documento ERS.

Algunas características pueden no necesitar la intervención del cliente; hay casos en los que los puede proponer el analista o, incluso, adoptarlos unilateralmente las decisiones y requisitos que el considere pertinentes (en características funcionales así también en las no funcionales). Como ejemplos tenemos: Algunas características relacionadas al tipo del sistema, requisitos no funcionales como

pueden ser los referentes al rendimiento, nivel de soporte a los errores operativos, plataformas para el desarrollo, interacciones o líneas de información (entre registros o tablas de datos) a almacenar en el caso de bancos o bases de datos, etc. También algunas funcionales como pueden ser características complementarias o de soporte necesarias para mejorar y facilitar el funcionamiento etc.

EL desarrollador obligadamente debe llegar a conocer sobre la tema y la problemática que se va a tratar, dominarlo al punto que el futuro programa a desarrollar lo exija. Por esta razón el desarrollador debe poseer una alta disposición para entender los problemas en gran cantidad de áreas o disciplinas de trabajo (que no suelen ser las de su dominio); así por ejemplo, si el programa a diseñar se enfoca en gestionar información de una aseguradora incluidas sus sucursales remotas, el analista se debe compenetrar en cómo la aseguradora trabaja y maneja su información, desde las niveles más bajos hasta incluso llegar a los niveles gerenciales. Dada la gran diversidad de campos de desarrollo que el analista debe cubrir, pueden recibir la colaboración de especialistas, es decir ayuda de profesionales que conoce a cabalidad el tema para el cual se diseñara el software.

#### *1.3.2.2. Diseño Del Sistema.*

Tiene como punto de partida las especificaciones de requisitos que se obtuvo en la fase de análisis, el diseño del sistema determina de qué manera estos requisitos se logran, la estructura que se le dará al sistema para que el software se haga realidad como se espera.

El diseño es una etapa completamente aislada de la programación, que es la traducción de las características y requerimientos en un determinado lenguaje de programación.

Las actividades hasta ahora nombradas no poseen distinciones evidentes cómo debería en las teorías de ingeniería de software.

Particularmente, el diseño, puede desarrollar el funcionamiento interno del programa en diferentes escalas de detalle, y cada uno de ellos se coloca en un punto medio entre el análisis y codificación.

Generalmente se refiere por "diseño de la arquitectura" al diseño de "muy alto nivel", que únicamente describe la estructura del programa en función de los módulos programados que lo componen y las relaciones entre estos módulos. A esta categoría de diseño corresponden fórmulas como cliente-servidor o "tres niveles", o, comúnmente, las decisiones relevantes al uso de la arquitectura de un hardware especial que se desee utilizar, así como el sistema operativo, DBMS, Protocolos de red, etc.

Un punto medio de detalle podría determinar la separación del sistema en módulos, pero incluyendo una observación más o menos específica al modo de separación que plantea el lenguaje de programación con el que el programa se llevara a cabo, como ejemplo tenemos, en un desarrollo llevado a cabo con la metodología de objetos, el proyecto tendría la posibilidad de ser definido en términos de clases y sus interrelaciones.

La estructura definida, al final, es una apreciación del programa próxima a la codificación (así pues, detallar no sólo las clases en, sino además sus cualidades y los métodos).

Debido a la naturaleza "intangible" del software, y dependiendo de las herramientas y características que se utilizan en el proceso, la frontera entre la codificación y el diseño también son prácticamente imposibles de identificar.

#### *1.3.2.3. Codificación Del Software.*

En esta fase de trabajo se llevan a cabo comúnmente las tareas que se conocen como codificación; las mismas que consisten en llevar a código fuente, en el lenguaje de programación seleccionado, todo el diseño de las etapas anteriores así como sus características. Esta tarea es llevada a cabo por el programador, guiándose con los lineamientos implantados en el diseño y tomando en cuenta siempre los requisitos funcionales y no funcionales (ERS) definidos en la primera fase del desarrollo.

Comúnmente se piensa que la etapa de programación o codificación es la que conlleva mayor esfuerzo y es la mayor parte del desarrollo del software; pero esto suele ser parcial ya que las fases anteriores son vitales y críticas e incluso pueden llevar más tiempo.

Mientras menor es el nivel del lenguaje que se usa para programar más largo será el tiempo de codificación necesario, así por ejemplo codificar un algoritmo en lenguaje ensamblador tardaría más tiempo que si programamos el mismo algoritmo en lenguaje C.

Durante la programación de la aplicación, sistema, o software generalmente se llevan a cabo también tareas de depuración, que significa ir liberando al código de los errores fáciles de ser hallados en esta fase, como pueden ser errores de semántica, sintáctica y lógica. Obviamente no todos los errores son detectados en la fase de codificación, siempre habrá más errores que se detectaran en el desarrollo las fases siguientes. La detección de algún error funcional (mala respuesta a los requerimientos) puede conllevar un retorno a la fase de diseño antes de seguir con la programación.

Durante la fase de programación, el código puede adoptar varios estados, según la forma de trabajo y el lenguaje de programación elegido, a saber:

- **Código fuente:** Es el código escrito personalmente por el programador en editores de texto, el cual genera su software. Este código contiene un conjunto de instrucciones codificadas en algún lenguaje de programación. Este código según las necesidades puede encontrarse distribuido en paquetes, procedimientos, bibliotecas fuente, etc.
- **Código objeto:** El código objeto no es comprensible directamente por el hombre este se encuentra comúnmente en formato binario, pero tampoco puede ser directamente ejecutable por el computador. Se trata de una encarnación media entre el código fuente y el código ejecutable, a los enlaces finales entre las rutinas de biblioteca y entre procedimientos o bien para ser utilizados con un pequeño programa que haga las veces de un intérprete intermedio.

Cuando el profesional usa un lenguaje a modo de Interpretador puro no existe este punto medio de código objeto, sino que por el contrario el mismo intérprete hace las veces de traductor y ejecuta el código fuente línea por línea siguiendo el flujo diseñado para el software. En este tipo de programación incluso no existen los archivos de código ejecutable. Uno de los mayores inconvenientes al momento de trabajar bajo este tipo de programación es que los tiempos de ejecución del software son mucho más largos que al utilizar un intérprete intermedio o si se basara en archivos de código ejecutable. Por lo cual no beneficia al rendimiento concerniente a velocidades de ejecución. Pero en contra parte posee una ventaja muy grande el trabajar a modo de intérprete puro, y es que este tipo de programación facilita en gran medida la eliminación de errores en el código fuente. Usualmente se suele trabajar bajo una forma mixta (siempre y cuando así lo permita el lenguaje de programación seleccionado), es decir se inicia el trabajo a modo de intérprete puro, y cuando sean eliminados los errores en el código fuente se obtiene el código ejecutable mediante la utilización de un compilador del lenguaje utilizado en la programación, trabajando mediante este tipo de programación mixta se consigue una eliminación de errores más rápida y se logra una optimización en las velocidades de ejecución del programa.

- **Código ejecutable:** Es el código binario resultante al enlazar uno o más fragmentos de código objeto con las rutinas y bibliotecas necesarias. Constituye uno o más archivos binarios con un formato específico que el sistema operativo es apto para cargarlo en la memoria RAM (eventualmente también se cargara en parte de la memoria virtual), y proceder a su ejecución directa.

#### 1.3.2.4. *Pruebas (Unitarias Y De Integración).*

Al software se le realizan una serie muy variada de pruebas entre las que se puede señalar las siguientes:

- **Prueba unitarias:** son aquellas que consisten en comprobar o testear las piezas de pequeñas del programa; a nivel de secciones, procedimientos sencillos, funciones únicas y módulos aislados; y todo tipo de elementos que realicen una función determinada dentro del programa. Dichas pruebas se utilizan para asegurar que estas secciones pequeñas o específicas del código estén en

correcto funcionamiento y que realizan sus funciones concretas dentro del conjunto con cierto grado de independencia.

- **Pruebas de integración:** estas pruebas se realizan una vez comprobadas exitosamente las pruebas unitarias; con éstas se busca comprobar el correcto funcionamiento del sistema completo, incluso los subsistemas que lo componen, las piezas individuales grandes que conforman el programa, funcionando adecuadamente al operar e interoperar como un todo.

Estas pruebas por lo general se realizan con un conjunto seleccionado de datos típicos a los que podría exponerse el programa, los módulos o los bloques de código.

#### *1.3.2.5. Instalación Y Paso A Producción.*

Se denomina instalación de un programa al proceso por medio del que los sistemas diseñados así como sus respectivos datos y configuraciones internas de funcionamiento son transferidos correctamente en computador de destino, inicializados, y, eventualmente, configurados por el usuario; todo ello con el único objetivo de poder ser utilizados finalmente por el usuario o cliente. Constituye así la etapa final dentro del desarrollo del programa. Luego de ésta etapa el software pasara a la etapa de funcionamiento y producción, para el que ha sido desarrollado.

La manera en que se llevara la instalación, depende de la manera en que el sistema fue desarrollado, y puede consistir desde una sencilla copia del medio en el que se almacena el programa hacia el disco de destino (algo que actualmente ya no se da); o más común, con una instalación de dificultad media en la cual los distintos archivos que componen el software (ejecutables, bibliotecas, datos propios, etc.) son descomprimidos y copiados en sitios determinados, previamente establecidos dentro del sistema y del disco; incluso pueden crearse relaciones con otros programas, no siempre únicamente con propio sistema operativo. Este caso de instalación media, por lo general es un proceso automatizado casi en su totalidad, desarrollado y guiado con herramientas del software específicas (empaquetado y distribución, instaladores).

Ya instalado el programa de manera satisfactoria, este pasa a la llamada etapa de funcionamiento, operatividad o producción, en la misma que desempeña las funciones para las que fue diseñado y desarrollado el software, es decir, es utilizado finalmente por el usuario o usuarios finales, para producir los resultados esperados.

### **1.3.3. Fases del Desarrollo del Software**

El término fases de desarrollo se usa para referirse a las diferentes etapas de progreso de desarrollo de un software y cuánto desarrollo puede requerir. Todo producto tiene diferentes versiones cada una de ellas se debe al paso de las diferentes etapas, todo software pasa por lo general por una fase durante la cual se insertan las nuevas características (etapa alfa), posteriormente pasa a una fase donde se depuran los errores de fácil detección (etapa beta), y por último pasa a una fase en donde se han retirado o reparado todos los errores relevantes (etapa estable). Hay etapas intermedias que también pueden ser reconocidas aunque no necesariamente se lo haga. Las etapas se pueden definir y regular adecuadamente por los desarrolladores del producto, pero los términos usados para definir las diferentes versiones se usan por lo general, de un modo poco formal para referirse al punto de desarrollo en el que se encuentra un programa. Es muy común encontrar que varios desarrolladores utilicen nombres de pila para los diferentes estados de un programa antes de ser lanzado, aun cuando las características o el programa en sí no son un secreto.

- **ALFA/ALPHA**

Es el programa en su primer estado, y es necesario que sea enviado a los verificadores para que sea probada.

Algunos equipos de desarrollo usan la denominación alfa informalmente para referirse a un producto todavía inestable, que aguarda aun una revisión la cual sirve para suprimir los posibles errores o para poner a prueba toda la funcionalidad del programa, pero que a pesar de estado ya satisface la mayoría de los requisitos generales.

- **BETA**

Se nombra como beta a la primera versión completa de un software, que aún puede encontrarse en un estado inestable pero que sin embargo sirve para las demostraciones internas y las inspecciones previas. Existen desarrolladores llaman a esta etapa como inspección previa (preview) o como una inspección previa técnica (technical preview [TP]). El software inicia esta fase, generalmente, después de que los desarrolladores fijan las características definitivas del software, lo cual indica que ya no se implementaran más características al programa y que únicamente se realizarán pequeñas correcciones o ediciones. Las versiones beta se encuentran en un punto medio en el flujo de producción del software. Los desarrolladores realizan pruebas de usuario con un grupo de personas seleccionadas llamados comúnmente betatesters, es muy común que se incluya al público en general. Los probadores o betatesters hacen una retroalimentación a los desarrolladores informando de cualquier error que encuentran y además de ciertas características, a veces de menor importancia, que les gustaría ver en la versión final.

Como la nomenclatura usada para definir estas etapas está basada en el alfabeto griego y al ser esta la segunda etapa del ciclo se la denomina con la segunda letra del alfabeto griego por lo que es llamada versión “beta”.

- **VERSIÓN CANDIDATA A DEFINITIVA (RC)**

Un programa en su estado de candidato a definitiva, candidato a versión final o candidato para el lanzamiento, pero más conocida por su nombre en inglés release candidate, hace referencia a un programa finalizado, prácticamente listo para lanzarse como versión definitiva, siempre y cuando no se encuentren errores que no lo permitan. El programa en esta fase ya posee todas las funciones del diseño y se considera que se encuentra libre de cualquier error o bug que implique un punto muerto en el desarrollo. Muchos desarrolladores utilizan frecuentemente este término. Otros de los términos relacionados a esta etapa incluyen gamma y delta para versiones que están prácticamente completas pero que aún se encuentran en pruebas; y omega para programas que se considera que están exentos de errores y se encuentran sometidos a la fase de pruebas

finales. Gamma, delta y omega son, respectivamente, la tercera, cuarta y última letras del alfabeto griego.

- **VERSIÓN DE DISPONIBILIDAD GL (RTM)**

La versión de disponibilidad general conocida también como "dorada" de un programa, es una versión generalmente idéntica a su versión previa, la candidata a definitiva, a la cual se le han realizado modificaciones de último momento. Esta versión es considerada muy estable y prácticamente libre de errores que posee una calidad adecuada para la distribución amplia y para ser usada por los usuarios finales. Las versiones comerciales, se pueden encontrar firmadas, esto se usa para que el usuario final tenga la seguridad de que el programa no ha sido modificado desde su salida. La expresión de que un programa "se ha dorado" hace referencia de que el programa se ha concluido satisfactoriamente y que este pasara a ser producido a gran escala y para ser vendido.

El término "dorado" se usa anecdóticamente por uso del "disco maestro de oro" que fuera usado para enviar la versión final a los fabricantes que lo usaban para crear las copias que saldrían a la venta. Este término se considera más una herencia de la producción musical. Sin embargo existen casos en los que en realidad el disco maestro es de oro esto para asegurar que el producto llegue sin ningún daño debido a la resistencia que presenta a la corrosión y segundo por apariencia estética.

#### **1.3.4. Licencia de Software**

Una licencia de software es un compromiso obligatorio entre el licenciante que puede ser el autor del software, el titular de los derechos de explotación o el distribuidor a quien se le atribuyen derechos especiales; y el licenciataro del programa informático que viene a ser el usuario o consumidor ya sea este una persona natural, un profesional o una empresa.

Esta licencia permite usar el programa sometido a una cantidad de términos y condiciones definidos en sus cláusulas.

Las licencias de un software pueden definir varias condiciones entre las que tenemos: la cesión de derechos específicos del propietario al usuario final sobre el software, el margen de responsabilidad sobre los fallos, el plazo durante el cual licenciatario gozara de los derechos que le provee la licencia, el margen geográfico efectivo del contrato además la licencia puede fijar varios compromisos del usuario final con el propietario, tales como la prohibición de prestar el programa a terceros o prohibir la instalación del programa en equipos deferentes al que se instaló inicialmente.

#### *1.3.4.1. Partes De Una Licencia De Software.*

##### **Licenciante**

El licenciante o la persona o empresa que suministra tanto del programa como de su licencia de uso al licenciatario, con la cual, el licenciatario tendrá determinados derechos sobre el programa. Se puede denominar como licenciante a cualquiera de los siguientes:

- **Autor:** Es el desarrollador o conjunto de desarrolladores que programan, desarrollan y crea el software son quienes por excelencia tienen en primer lugar el papel de licenciante al ser los titulares originales del programa.
- **Titular de los derechos de explotación:** Es la persona natural o jurídica al que se le otorga una cesión exclusiva de los derechos de explotación del programa, transformándolo en un dueño derivado y licenciante del programa.
- **Distribuidor:** Es la persona jurídica a la cual se le cede un privilegio especial para la distribución y además la facultad para crear sublicencias del programa por medio de un acuerdo de distribución con el titular de los derechos de explotación.

##### **Garantía de Titularidad**

Es un respaldo que otorga el licenciante o propietario, mediante el que se confirma que tiene con los privilegios indispensables para el aprovechamiento del programa como para autorizarle de entregar una licencia al licenciatario.

##### **Licenciatario**

Mejor conocido como usuarios, es una persona física o jurídica a la cual se le autoriza el uso del programa, más uno que otro privilegio para el aprovechamiento del programa, siempre y cuando cumpla con las normas descritas en el acuerdo.

- **Usuario consumidor:** es una persona natural a la que le haya entregado una licencia del programa cedida por el licenciante. El usuario estará en desventaja ante los términos y condiciones descritos en ella ya que debe aceptarlos obligatoriamente al momento de adquirir el software sin ningún tipo de atribución especial.
- **Usuario profesional o empresa:** es una persona natural o jurídica a la que se le concede una licencia del programa cedida por el licenciante, en la que se encuentra en condiciones iguales al licenciante para usar sus privilegios y deberes ante los términos y condiciones descritos en el contrato.

### **Plazo**

El plazo es el que define el periodo durante el que serán válidos los términos y condiciones descritos en el contrato. Según el plazo los contratos o licencias pueden ser clasificados en:

- Licencias de plazo específico.
- Licencias de plazo indefinido.
- Licencias sin plazo específico.

### **Precio**

El precio establece el valor que deberá ser cancelado por el licenciatarario al licenciante ya sea para el uso del programa o para la adquisición de los privilegios descritos en el contrato.

#### *1.3.4.2. Clasificación De Las Licencias De Software.*

### **Licencia de software de código abierto permisivas**

Da la posibilidad de producir programas derivados del software original sin que ésta tenga alguna obligación o derecho. Muchas licencias siguen este lineamiento.

### **Licencias de software de código abierto robustas fuertes**

Las licencias con copyleft fuerte, generalmente poseen estipulaciones en las que se obliga a que cualquier modificación que se hagan al software o que cualquier software derivado de producto original deban ser licenciadas bajo los mismos términos y condiciones de la licencia original.

### **Licencias de software de código abierto robustas débiles**

Las licencias copyleft débil o también las licencias híbridas, suelen contener estipulaciones en la que se exige a que cualquier modificación que se realicen al programa original se deba licenciar bajo los mismos términos y condiciones que los de la licencia original, pero con la notable excepción de que cualquiera de las obras derivadas del software original puedan ser licenciadas bajo sus propios términos y condiciones distintos al de la licencia original.

### **Licencia de software de código cerrado**

A este tipo de contratos también se los llama con el nombre de software propietario o privativo. En este tipo de licencias el propietario define los atributos de uso, distribución, redistribución, copia, modificación, cesión y todo tipo de contemplaciones el propietario valore como necesarias o indispensables.

Este tipo de licencias, generalmente, prohíbe que el software sufra cualquier tipo de modificación, desensamblado, copiado o distribución que no sea especificada bajo la propia licencia, a las modificaciones que incumplan estos lineamientos es a lo que se denomina normalmente como piratería de software.

Este tipo de licencias también regula el número de copias que pueden ser instaladas e inclusive puede llegar a establecer regulaciones sobre los fines concretos para los cuales se puede utilizar el software.

Por lo general este tipo de contratos restringen en gran medida la responsabilidad sobre los fallos del software.

Los fabricantes de programas que se regulan con este estándar de contratos comúnmente brindan soporte técnico así como servicio de actualizaciones frecuentes durante el tiempo de vida del producto o más comúnmente durante el tiempo de vigencia de la licencia.

### **Software de dominio público (sin licencia)**

En este tipo de software se permite de manera libre cualquier tipo de uso, copia, modificación o redistribución sea con fines de lucro o no.

## **Licencia de Usuario Final**

En inglés End User License Agreement por sus siglas más conocido como EULA, es un tipo de licencia mediante la cual el uso del producto está permitido exclusivamente a un único usuario que viene a ser el comprador.

En este caso, la licencia detallada por el dueño de los derechos del software exige al usuario final del producto a que reconozca tener conocimiento de las condiciones y derechos para su uso, así como de las patentes, etc. y que acepte de conformidad a todo ello.

El conocimiento de todas las condiciones de los contratos es difícil antes de comprar un producto ya que generalmente las cajas de los productos no cuentan con una copia detallada de las mismas, por lo que el comprador en la gran mayoría de las ocasiones desconoce su contenido hasta después de la compra. En muchas ocasiones además se exige a que el usuario renuncie a su derecho de realizar reclamos o demandas legales por varios motivos como podrían ser fallos o daños en el producto, o a aceptar la restitución del producto en la forma o monto que el licenciante considere adecuado para ambas partes.

Este tipo de acuerdo manifiesta además que tipo de uso se le pueden dar al software y, así mismo, cuáles no; ya que el comprador no es, legalmente, bajo ninguna forma dueño del producto, sino únicamente se lo considera como un poseedor de una licencia para su uso, por lo cual algunas personas consideran este hecho como una restricción a sus derechos como consumidor.

Este tipo de acuerdos son unilaterales ya que el usuario no tiene ninguna opción más que aceptar o rechazar el contenido del mismo que equivale a adquirir o no el producto, existe en algunos países organizaciones dedicadas a la protección del consumidor que autorizan ciertas excepciones a estas cláusulas.

## **Licencia de distribuidores**

Bajo este tipo de licencia, se le proveen atribuciones específicas a un comerciante de tipo comisionario para que venda o distribuya el software dándole una parte de la ganancia al fabricante.

Este tipo de licencia en ninguna instancia se refiere a una licencia de uso en términos jurídicos, sino más bien equivale a un acuerdo comercial en la cual no necesariamente se deben ceder los derechos para la distribución. Muchas veces se da el caso de una sencilla actividad de comercio en la cual inclusive el distribuidor no tiene siquiera contacto con el producto, en el cual el producto así como el contrato de uso en si son suscritas y puestas a disposición por el fabricante, encargándose como un intermediario el distribuidor únicamente del cobro al usuario y del pago al fabricante con su respectivo margen de ganancia.

## CAPITULO II

### 2. MATERIALES Y MÉTODOS

#### 2.1. Materiales Y Equipos

- Computador de Escritorio
- Computador Portátil
- Visual Studio 2012
- Flash Memory

#### 2.2. Métodos

##### *2.2.1. Método de Selección de Procesos*

Para facilitar y optimizar el desarrollo del proyecto se tuvo que seleccionar los procesos de tratamiento que estarían presentes dentro del proyecto, con este objetivo se hizo una selección por criterios buscando los procesos que cumplan con los criterios de selección establecidos:

**Popularidad:** criterio usado para estableceres que proceso es usado habitualmente o que proceso tiene mayor difusión de uso.

**Economía:** criterio usado para determinar los procesos que económicamente son más viables y por lo tanto pueden tener mayor difusión.

**Construcción:** criterio usado para determinar los procesos que representes una mayor facilidad de construcción e implementación.

**Operación:** criterio usado para determinar los procesos que sean más fáciles de operar una vez implementados.

**Mantenimiento:** criterio usado para determinar los procesos que presente un mantenimiento más fácil.

**Cálculos:** criterio usado para determinar los procesos menor dificultad de cálculo para facilitar la programación.

### **2.2.2. Método para el Diseño de dos Procesos**

Para llevar a cabo el diseño y dimensionamiento de los diferentes procesos seleccionados de una planta de tratamiento de aguas se tomó como metodología los diferentes criterios y ecuaciones presentes en las bibliografías de la investigación.

### **2.2.3. Metodología para la Programación**

El desarrollo del software tuvo como base la metodología de desarrollo ágil XP aplicando el paradigma de programación orientado a objetos.

La metodología de desarrollo ágil XP es una metodología que se basa en impulsar las relaciones interpersonales, fomentando el trabajo en grupo, procurando un aprendizaje en los desarrolladores. Se basa en la realimentación continua entre el cliente y el equipo de desarrollo por lo cual es una metodología capaz de adaptarse a los cambios de requisitos en cualquier punto de desarrollo del proyecto.

La programación orientada a objetos es un paradigma de la programación en el cual se usan los objetos sus interacciones para crear un programa, en el cual pensamos en un objetos y luego definimos sus atributos o características lo cual lleva a programar la estructura y función del objeto dentro del programa.

### **2.2.4. Metodología de Validación de la Herramienta**

Para la comprobación del cumplimiento de los objetivos del software tanto para sus funciones y características, así como para la fiabilidad de los resultados, se sometió el programa a una calificación usando el "Método Delphi"; en el cual se facilitó el software a un grupo de profesionales para que lo usen y hagan las pruebas que ellos consideren necesarias, para posteriormente llenar un cuestionario que permita calificar las diferentes características del software y que también permita recibir una retro alimentación de las consideraciones o sugerencias que puedan tener los profesionales.

Además de esto se usó “La Escala de Likert”, para poder obtener valores totales de en el cumplimiento de los objetivos del software; siendo 24 el máximo puntaje equivalente a un 100% de satisfacción al momento de usarlo.

De esta manera se puede obtener resultados tanto cualitativos como cuantitativos del proyecto finalizado, a la vez que permite obtener conclusiones y recomendaciones desde varios puntos de vista.

El método Delphi es un método de estructuración para un proceso de comunicación grupal el cual es efectivo al momento de permitir que un grupo de individuos trabaje como un todo para tratar un problema complejo.

Entre las características más destacadas al momento de usar este método tenemos:

**Anonimato**, debido a que durante el desarrollo del proceso ninguno de los expertos conoce la identidad del resto de expertos.

**Retroalimentación**, ya que la aplicación de este método permite obtener datos específicos según la encuesta, así como datos aleatorios de las opiniones personales de los expertos.

**Resultados Estadísticos**, los resultados son fácilmente procesables de manera estadística ya que estos son arrojados a manera de “grado de Acuerdo”, y permiten un análisis estadístico.

**Heterogeneidad**, ya que este método permite la participación de expertos de determinadas ramas de una actividad sobre las mismas bases.

Entre las ventajas de la aplicación de este método tenemos: permite obtener información de los puntos de vista de temas ya sea muy amplios o muy específicos, el horizonte al que se lleve el análisis puede ser muy variado y depende de los resultados que se desee obtener, permite la participación de grupos de personas ya sean grupos pequeños o grupos grandes sin que el manejo del grupo se convierta en un problema, explorar de forma sistemática y objetiva problemas que requieren de opinión cualificada.

Las desventajas del uso de este método son: el largo tiempo de ejecución, sesgos en la correcta elección de los participantes, elevado número de deserciones.

## CAPITULO III

### 3. DESARROLLO Y RESULTADOS

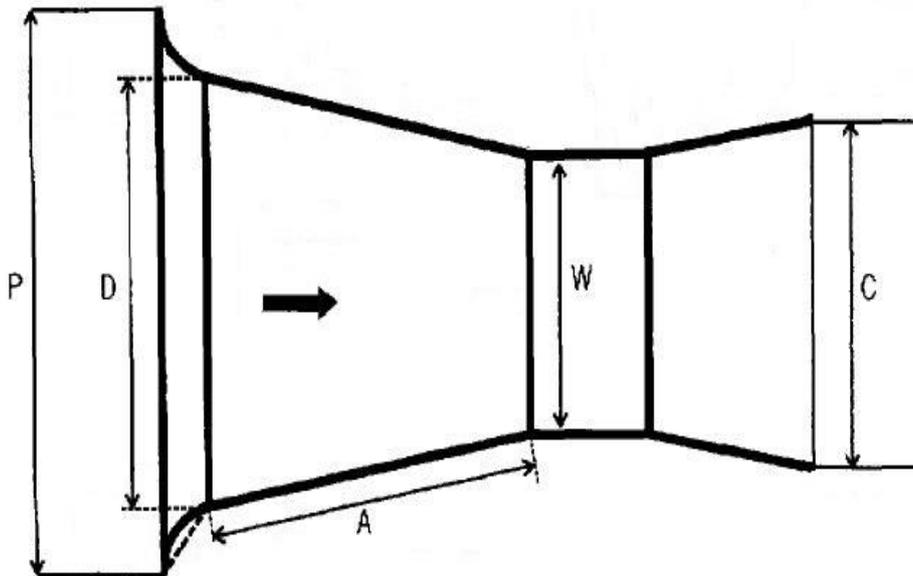
#### 3.1. Desarrollo de los Procesos de Tratamiento

##### 3.1.1. Desarrollo del Canal Parshall

Es una estructura de diseño calibrado, de forma abierta tiene una sección convergente, una garganta y una sección divergente.

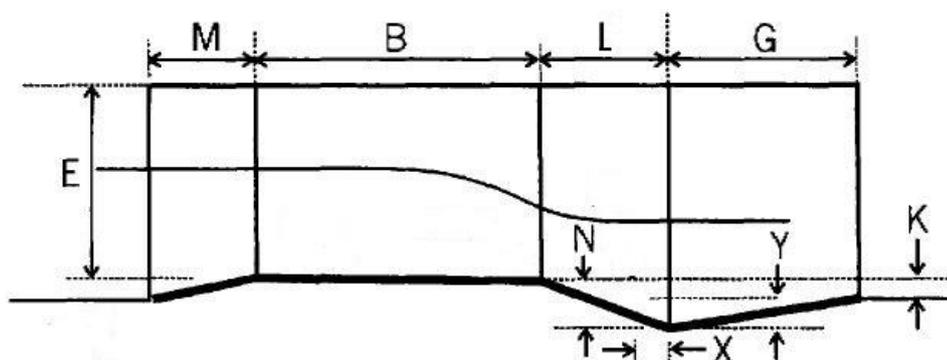
Los usos del canal parshall son: reducir la velocidad de entrada del flujo sin importar la velocidad en la que ingrese al canal, constituye una primera etapa de sedimentación, sirve como estructura de medición y tiene la ventaja de operar en un rango amplio de flujos.

**Ilustración 1:** Vista Planta del Canal Parshall



Fuente: Internet

**Ilustración 2:** Vista Lateral del Canal Parshal



Fuente: Internet

Como se observa en las gráficas se deben asignar diversos valores los mismos que están contenidos en tablas (anexos) en función al Caudal "Q", el valor del caudal para el cálculo del procesos será ingresado por el usuario en l/s. Si el caudal supera el valor máximo de las tablas se deberá proceder a dividir el caudal calculara dos Parshall para el caudal dividido.

### **3.1.2. Desarrollo de las Rejillas**

Las aguas residuales frecuentemente contienen materiales solidos de gran tamaño como cáscaras de fruta, harapos, palos, trozos de papel y madera que encuentran su destino en el sistema de alcantarillado, estos residuos se deben separar del agua, esto se logra pasando las aguas a través de rejillas, hechas comúnmente con varillas de hierro paralelas.

El principal propósito de estos dispositivos de cribado es el proteger toda clase de equipos electromecánicos que puedan ser requeridos en el proceso de tratamiento y prevenir el atascamiento de válvulas. Por este motivo el cribado es la primera operación que se lleva a cabo en el proceso de tratamiento.

La estructura funcional de las rejillas, deben ser colocadas inclinadas con respecto al suelo.

Las rejas de limpieza manual se instalan con una inclinación de 30 a 60° con respecto suelo sobre el cual se instalan, en cambio, las rejillas de limpieza mecánica poseen una inclinación mucho mayor, que va de 60 a 90°.

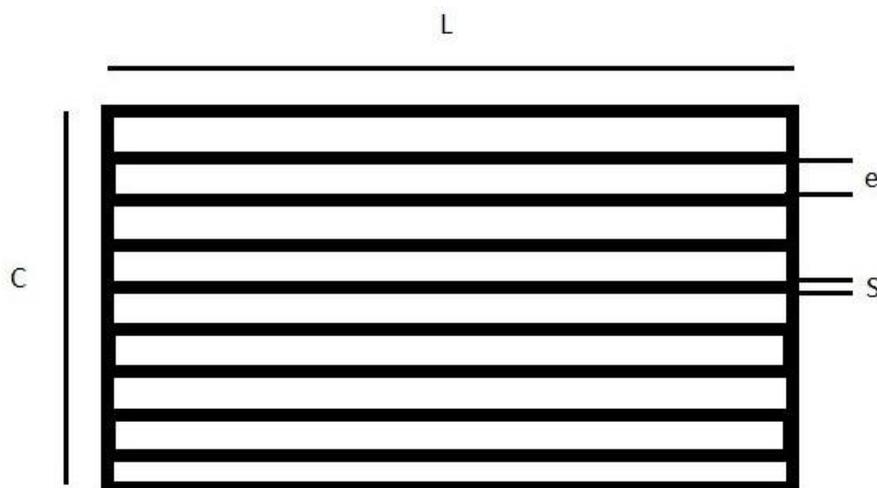
Las barras de las rejillas pueden tener diversas formas aunque comúnmente se usan las rectas o planas y estas debe estar separadas existiendo así las de separación grande, de 5 a 10 cm, y las de separación pequeña, de 2 a 4 cm.

A medida que las rejillas filtran el agua de ingreso comienza a acumularse el material retenido esto produce un aumento en el nivel del agua en el canal de llegada, se debe evitar una acumulación excesiva de material retenido ya que esto impide que las partículas de menor tamaño que la separación entre barras no puedan pasar a través de ellas lo cual no es conveniente para procesos posteriores.

Por lo general el flujo en el canal debe ser laminar, sin embargo se recomienda que el agua tenga por lo menos una velocidad de 0.5 m/s para así detener los materiales grandes, dejando pasar las partículas pequeñas. Pero durante la época de lluvia cuando la velocidad se incrementa; es recomendable que bajo estas condiciones la velocidad máxima sea de 2.0 m/s.

Cuando no es posible estar dentro del margen de velocidad recomendado debido al ancho requerido, es altamente recomendable diseñar dos canales independientes con la finalidad de dividir el gasto de agua y de esta manera disminuir la velocidad de flujo en tiempo de lluvia.

**Ilustración 3:** Modelo ilustrativo de una Rejilla



Autor: Cristian Suárez

El caudal “Q” debe ser ingresado por el usuario, este estará en m<sup>3</sup>/s.

Área (A)

$$A = \frac{Q}{V(\text{velocidad del agua})}$$

La velocidad “V” lo tomaremos como 0.6m/s que es el valor máximo para este tipo de rejillas.

Altura de la lámina de agua (h)

$$h = \frac{A}{C}$$

El dato (C) que es el ancho del canal, deberá ser ingresado por el usuario y deberá estar en m.

Longitud de la rejilla (L)

$$L = \frac{h}{\text{Sen } \emptyset}, \text{ L estará en metros.}$$

$\emptyset$  corresponde al ángulo de inclinación de la rejilla y debe ser asignado por el usuario.

Numero de barras requeridas (n)

$$C = (n * S) + ((n - 1) * e)$$

$$n = \frac{C + e}{S + e}$$

Donde “C” es el ancho del canal que para este cálculo se trasformara en centímetros, “S” es el espesor de las barras que deberá ingresar el usuario en centímetros y “e” es el espacio entre barras pudiendo ser este de 5 a 10cm en rejillas de separación grande y de 2 a 5cm en rejillas de separación pequeña.

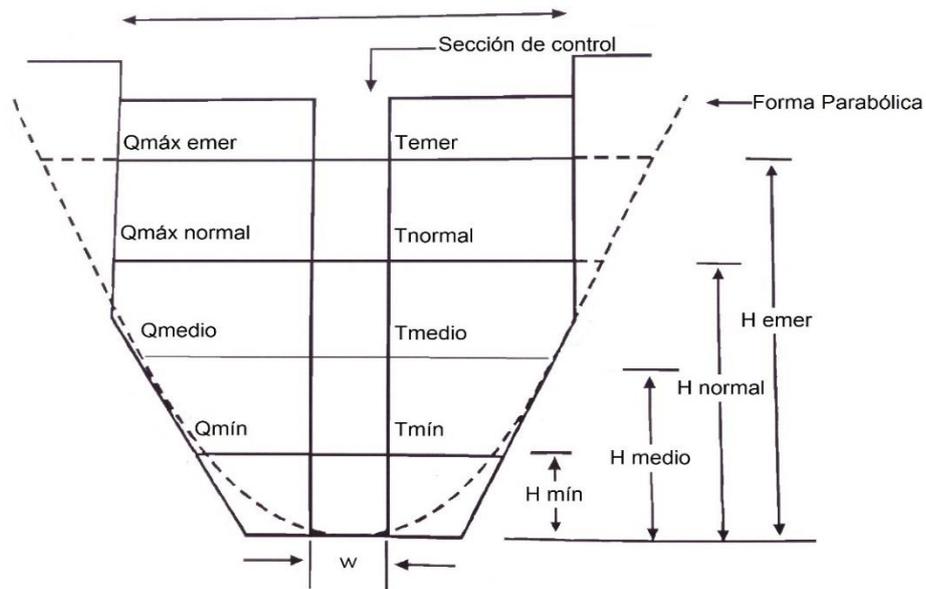
### **3.1.3. Desarrollo del Desarenador Horizontal**

El procedimiento de separación de la arena consiste en la reducción de la velocidad del agua, bajo los límites de precipitación pero sobre los de sedimentación de la materia orgánica presente dado que la materia orgánica será necesaria para los procesos que se realizaran más adelante.

El principio básico consiste en provocar condiciones en las cuales el sólido de alta densidad pueda decantar. Esto se logra disminuyendo la velocidad del líquido.

Lo normal son velocidades  $\leq 0.3$  m/s, para separar arena con un diámetro superior a 0.2 mm.

#### Ilustración 4: Vista frontal de un Desarenador Horizontal



Autor: Cristian Suárez

Para este tipo de diseño se necesita que el usuario ingrese los datos siguientes datos:

- T normal (ancho del canal), sino ingresa se puede tomar el valor de "C" de las rejillas o el Aforador Parshall y debe estar en metros.
- V = Velocidad (la cual se tomara como 0.3 m/s que es lo normal para este tipo de desarenadores)
- Qmáx, en m<sup>3</sup>/s.
- Qmed, en m<sup>3</sup>/s.
- Qmínimo, en m<sup>3</sup>/s.
- Número de Canales (mínimo 2 canales).

Una vez ingresado los valores de Q se procede a calcular los valores de diseño de la siguiente manera:

- $Q_{\text{máx normal}} = Q_{\text{máx}} / \text{Número de canales}$
- $Q_{\text{medio}} = Q_{\text{med}} / \text{Número de canales}$
- $Q_{\text{mín}} = Q_{\text{mínimo}} / \text{Número de canales}$

- $Q_{\text{máx emer}} = Q_{\text{máx normal}} + Q_{\text{medio}}$ , en caso de que el número de canales no haya sido ingresado y se utilice el número de canales por defecto "2".  
 $Q_{\text{máx emer}} = Q_{\text{máx normal}} \times 2$ .

Con todos estos valores se procede a calcular las medidas de las diferentes secciones comenzando para la sección de  $Q_{\text{máx normal}}$ :

- $H_{\text{normal}} = \frac{3 \cdot Q_{\text{máx normal}}}{2 \cdot T_{\text{normal}} \cdot V}$
- $\frac{V_c^2}{2g} = \frac{1}{3.1} \left( H_{\text{normal}} + \frac{V^2}{2 \cdot g} \right)$
- $dc = 2 \frac{V_c^2}{2 \cdot g}$
- $V_c = \sqrt[2]{2 \cdot g \cdot \frac{dc}{2}}$
- $a = \frac{Q_{\text{máx normal}}}{V_c}$
- $W = \frac{a}{dc}$

Luego se procede a los cálculos para  $Q_{\text{medio}}$ :

- $a_{\text{medio}} = \sqrt[3]{\frac{Q_{\text{medio}}^2 \cdot W}{g}}$
- $dc_{\text{medio}} = \frac{a_{\text{medio}}}{W}$
- $H_{\text{medio}} = \frac{3.1 \cdot dc_{\text{medio}}}{2} - \frac{V^2}{2 \cdot g}$
- $T_{\text{medio}} = \frac{3 \cdot Q_{\text{medio}}}{2 \cdot H_{\text{medio}} \cdot V}$

Los cálculos para  $Q_{\text{min}}$ :

- $a_{\text{min}} = \sqrt[3]{\frac{Q_{\text{min}}^2 \cdot W}{g}}$
- $dc_{\text{min}} = \frac{a_{\text{min}}}{W}$
- $H_{\text{min}} = \frac{3.1 \cdot dc_{\text{min}}}{2} - \frac{V^2}{2 \cdot g}$
- $T_{\text{min}} = \frac{3 \cdot Q_{\text{min}}}{2 \cdot H_{\text{min}} \cdot V}$

Nos aseguramos de hacer los cálculos de emergencia al final con la  $Q_{\text{máx emer}}$ :

- $a_{emer} = \sqrt[3]{\frac{Q_{m\acute{a}x} \cdot emer^2 \cdot W}{g}}$
- $dc_{emer} = \frac{a_{emer}}{W}$
- $H_{emer} = \frac{3.1 \cdot dc_{emer}}{2} - \frac{V^2}{2 \cdot g}$
- $T_{emer} = \frac{3 \cdot Q_{m\acute{a}x} \cdot emer}{2 \cdot H_{emer} \cdot V}$

Y finalmente procedemos con los cálculos para la longitud de los canales:

- $L = \frac{H_{normal} \cdot V}{V_s}$ , donde “Vs” es la velocidad de sedimentación y es equivalente a 0.01916 m/s.
- $L_{m\acute{i}n \text{ adicional}} = 2 \cdot H_{m\acute{a}x \text{ emer}}$
- $L_{m\acute{a}x \text{ adicional}} = 0.5 \cdot L$

#### **3.1.4. Desarrollo del Filtro Anaerobio de Flujo Ascendente “FAFA”**

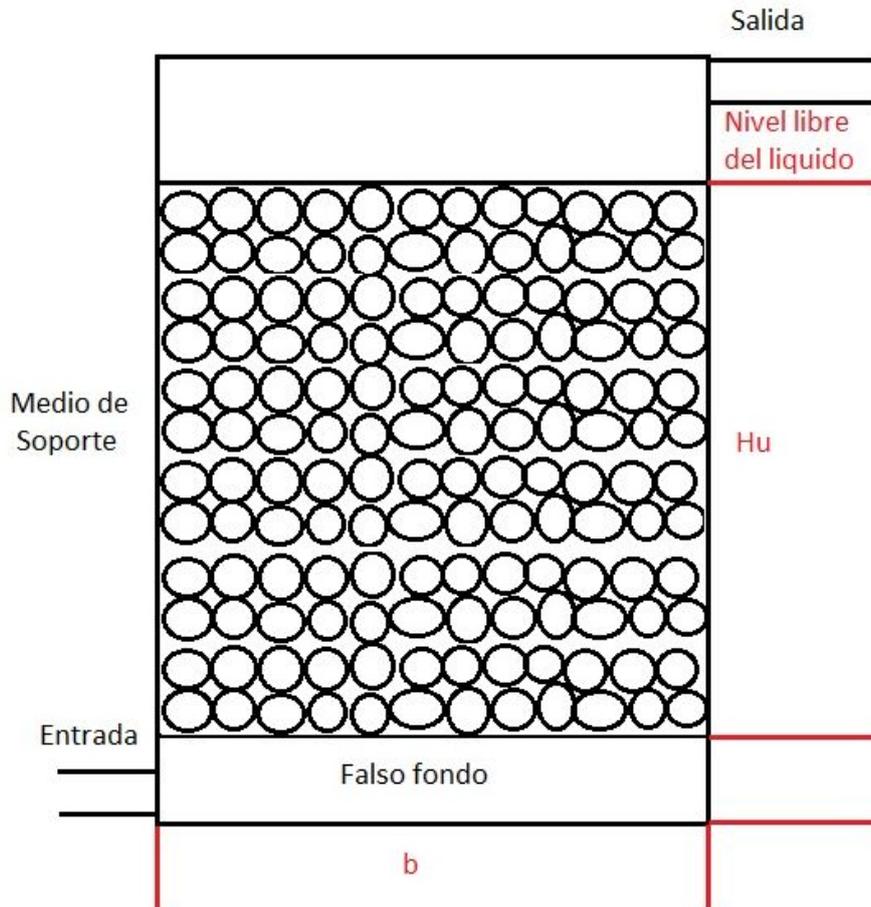
El Filtro Anaerobio De Flujo Ascendente o también denominado FAFA por sus siglas, es básicamente un filtro de grandes dimensiones el cual realiza tanto la función de filtrado como una función de degradación biológica, este proceso de degradación es de tipo adherido; es decir los microorganismos no estarán suspendidas en el agua sino que se encontraran adheridas al material del filtro (cuarzo, granito, bloques cerámicos o de PVC, esferas de polietileno, bambú, etc.; de granulometría uniforme con diámetros de 4-7 cm).

En este tipo de filtros se ha observado una eficiencia de hasta el 80% en la remoción de DBO; este porcentaje varía según factores como la temperatura, el caudal, el tiempo de retención hidráulica, las dimensiones del material del filtro, etc.

Este tipo de filtros además deben ser construidos en pares al igual que la mayoría de los sistemas de una planta de tratamiento de agua debido al mantenimiento que se les debe dar, que en el caso de estos filtros es un poco más largo que el de otros sistemas.

Además al construirlos se debe asegurar que el fondo falso perforado permita un adecuado ingreso y un flujo uniforme.

**Ilustración 5:** Vista lateral de un Tanque FAFA



Autor: Cristian Suárez

El caudal  $Q$ , será ingresado por el usuario y deberá estar en  $m^3/día$ .  
 Ancho del tanque ( $b$ ) será un valor asumido por el usuario, en metros.  
 El tiempo de retención hidráulica ( $tr$ ) debe ser ingresado por el usuario, en días.  
 La altura útil del filtro ( $Hu$ ), por norma se recomienda un valor de 1.8m.  
 Falso fondo, con una altura de 0.3m y el nivel libre del líquido antes de la descarga de igual manera 0.3m

- Volumen del filtro ( $V_f$ )

$V_f = \frac{Q}{\# \text{ de filtros}} * tr$ , el número de filtros debe ser ingresado por el usuario según sus necesidades de caudal.

- Área del filtro ( $A_f$ )

$$A_f = \frac{V_f}{H_u}$$

- Longitud del filtro ( $L$ ), en metros.

$$L = \frac{A_f}{b}$$

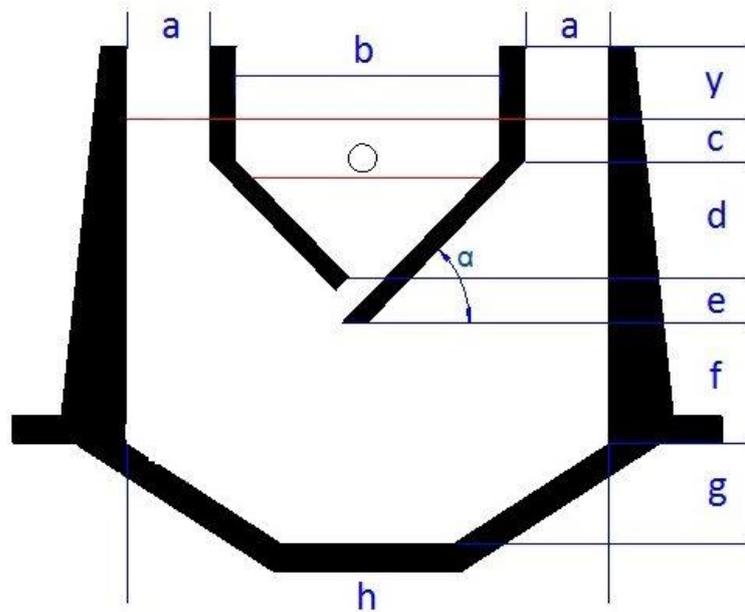
### **3.1.5. Desarrollo del Tanque Imhoff**

El tanque IMHOFF es un tanque dividido en cámaras las mismas que desempeñan funciones específicas; una cámara superior la cual por su forma especial cumple una función de sedimentación y a su vez una función de desviación de los gases producidos en la cámara inferior, una cámara inferior en la cual se depositan todos los sólidos sedimentados y en esta se degradan la materia orgánica.

Además de su diseño especial este tipo de tanques debe ser instalado con un sistema de drenado por bombas ya que el material sedimentado y degradado en la cámara inferior se acumula lo cual puede disminuir la eficiencia por lo cual esta cámara posee un fondo en punta o cono según la forma del tanque para en este punto colocar las tuberías que drenaran los lodos y que además de esta manera se asegura que se drenaran los lodos más viejos y ya degradados que se acumulan al fondo dejando así la capa de los lodos jóvenes para que continúen el proceso de degradación.

Para inicial el funcionamiento de estas cámaras es necesario agregar lodos activos al inicio de sus operaciones para asegurar el proceso de biodegradación o a su vez agregar nutrientes para que las bacterias presentes en el agua comiencen este proceso.

### Ilustración 6: Vista frontal de un Tanque Imhoff



Fuente: Internet

El caudal será ingresado por el usuario en m<sup>3</sup>/día.

Población (p), al ser estas plantas diseñadas según la población y la proyección de los años el usuario deberá ingresar el valor de la población.

Tiempo de retención Hidráulica (trh), debe ser ingresado por el usuario su rango esta entre 2 – 4 h.

Carga Superficial (Cs) debe ser ingresado por el usuario en m<sup>3</sup>/día y dentro del rango de 24 – 50.

Dotación de Lodos (Dlodos), este valor también lo deberá ingresar el usuario en un rango de 0.07 – 0.1 m<sup>3</sup>/habitante\*año.

Ancho del Sedimentador (b), deberá ser ingresado por el usuario y deberá ser un valor en metros.

El valor de “a” puede variar pero lo tomaremos como un valor constante en 0.6m.

El valor de “e” igualmente es variable dentro de un rango de 0.45 – 0.6m pero para facilidad lo tomaremos en el valor de 0.45m, a menos que el usuario desee ingresar un valor dentro del rango.

- Área del sedimentador

$$A_s = \frac{Q}{C_s}$$

- Longitud del tanque

$$L = \frac{As}{b}$$

- $d = \tan \alpha * \frac{b}{2}$

- $Vs = Q * (trh)$

- $At = \frac{Vs}{L}$

- $c = \frac{At - \frac{b*d}{2}}{b}$

- $h = 2a + b$

- $Vd = p * D\text{lodos}$

- $f = \frac{\left( vd - \frac{h^2 * L}{12} \right)}{h * L}$

- $g = \frac{h}{4}$

- Altura de seguridad (y), tomaremos como constante el valor 0.5m.

### **3.1.6. Desarrollo de la Laguna Anaerobia**

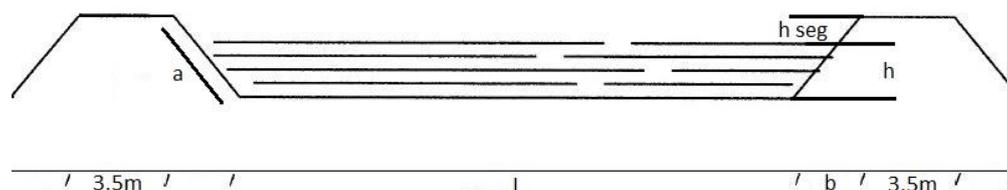
Los estanques anaerobios se usan para el tratamiento de agua residual de alto contenido orgánico que también contenga una alta concentración de sólidos.

Generalmente una laguna anaerobia es un estanque profundo excavado en el terreno, dotado de un sistema de conductos de entrada y de salida adecuados. Para conservar la energía calorífica y mantener las condiciones anaerobias, se construyen estanques de profundidades que varían entre los 2.4 m y los 9 m, siendo profundidades típicas entre 4 y 5 m.

Normalmente, estanques de este tipo de lagunas son anaerobios casi en su totalidad, únicamente en una delgada capa que limita con la superficie no posee esta característica anaerobia. La descontaminación en estas lagunas se basa en la precipitación de los contaminantes y en la degradación anaerobia de los residuos orgánicos produciendo CO<sub>2</sub>, CH<sub>4</sub>, otros productos gaseosos finales, ácidos orgánicos y tejido celular.

Normalmente en este tipo de lagunas se consigue fácilmente y de modo continuo, eficiencias de la eliminación de DBO<sub>5</sub> sobre el 70%. Incluso si se contara con condiciones óptimas de funcionamiento, se podría lograr un rendimiento en la eliminación de hasta el 85%.

### Ilustración 7: Vista lateral de una zanja para lagunaje



Autor: Cristian Suárez

La altura útil “h”, deberá ser ingresada por el usuario y el valor deberá dentro del rango de 3 – 6m.

La altura de seguridad “h seg” es un valor de altura adicional para evitar desbordamiento, este valor estará fijo en 0.3m.

El tiempo de retención “Rt”, deberá ser ingresado por el usuario y deberá estar dentro de un rango de 10 - 50 días.

El caudal “Q” deberá ser ingresado por el usuario y deberá estar en m<sup>3</sup>/día.

El ancho de la laguna “B”, deberá ser ingresado por el usuario en metros.

El ángulo de las paredes “θ”, deberá ser ingresado por el usuario, con la sugerencia de 45<sup>0</sup> que es el ángulo más utilizado.

- Volumen

$$V = Q * Rt$$

- Área

$$A = \frac{V}{h}$$

- Área de las lagunas

El área una vez obtenida será dividida para dos, ya que la mayoría de los procesos de las plantas de tratamiento se diseñan en pares.

$$Alag = \frac{A}{2}$$

- Longitud

$$L = \frac{Alag}{B}$$

- $b = \frac{h+hseg}{\tan\theta}$

- $a = \frac{h+hseg}{\text{sen}\theta}$

### **3.1.7. Desarrollo de la Laguna Aerobia**

Una laguna aerobia asegura condiciones aerobias en su totalidad, además, conteniendo bacterias y algas en suspensión. Se diferencian dos tipos principales de zanjas aerobias.

En las primeras, su finalidad es la de potenciar el desarrollo de algas. Este tipo de estanques suele limitar su profundidad de unos 15 a 50cm.

En los otros tipos de estanque, se observan profundidades de hasta 1.5m y su finalidad es promover el aumento de la cantidad de oxígeno producido. En estos dos tipos de lagunas aerobias, el oxígeno es producido por las algas pero también se obtiene de la atmósfera por difusión.

Para lograr mejores niveles de oxígeno, es prudente oxigenar los estanques usando bombas o aireadores de superficie.

En los estanques aerobios fotosintéticos, el oxígeno se suministra por aireación natural a través de la superficie y por fotosíntesis de las algas. Con excepción de la población de algas, la comunidad biológica presente en los estanques de estabilización es similar a la existente en los sistemas de fangos activados. El oxígeno liberado por las algas en el proceso de fotosíntesis es utilizado por las bacterias en la degradación aerobia de la materia orgánica. Los nutrientes y el dióxido de carbono liberados en este proceso de degradación los emplean, a su vez, las algas. También se presentan animales superiores tales como los rotíferos y protozoos, cuya principal función consiste en la mejora del efluente.

Al igual que la laguna anaerobia, la laguna aerobia se desarrolla bajo los mismos lineamientos y formulas, lo único que varía son los rangos de dos datos que debe ser ingresados por el usuario estos son:

- La altura útil "h", deberá ser ingresada por el usuario y el valor deberá dentro del rango de 0.6 – 1.2m.
- El tiempo de retención "Rt", deberá ser ingresado por el usuario y deberá estar dentro de un rango de 4 – 12 días.

### **3.1.8. Desarrollo de la Laguna Facultativa**

Los estanques facultativos llevan a cabo el proceso de descontaminación usando una combinación de bacterias facultativas, es decir que estas pueden

desarrollarse y actuar como anaerobias o aerobias según lo necesiten, a estas lagunas se les conoce también con el nombre de estanques de estabilización facultativas (aerobios- anaerobios).

En los estanques facultativos podemos diferenciar tres zonas muy marcadas:

1. Una zona en la que podemos encontrar algas así como bacterias aerobias, esta es la zona superficial del estanque.
2. Una zona en la que las bacterias anaerobias se encargan de descomponer todo tipo de material acumulado, esta es la zona inferior del estanque.
3. Y una zona que es parcialmente aerobia y anaerobia, en la que la descontaminación del agua es realizada por la acción de bacterias facultativas, esta zona es la intermedia del estanque.

En el fondo del estanque se acumulan los sólidos y material orgánico que han sedimentado para formar una capa que se conoce como fango anaeróbico. Los materiales orgánicos suspendidos o coloidales son degradados por las bacterias aerobias, así como también las facultativas, las cuales utilizan el oxígeno producido por las algas para oxidar la materia. Las algas a cambio obtienen su fuente de carbono del dióxido de carbono que se produce por el proceso de oxidación de las bacterias.

La descomposición anaerobia realizada en la zona del fango anaerobio en cambio produce compuestos orgánicos disueltos y gases tales como el CO<sub>2</sub>, el H<sub>2</sub>S y el SH<sub>4</sub>, que son liberados a la atmósfera a menos que sean oxidados por las bacterias aerobias.

De igual manera que las lagunas anaerobias y aerobias, las facultativas siguen los mismos lineamientos y fórmulas para su desarrollo con las siguientes consideraciones específicas:

- La altura útil "h", deberá ser ingresada por el usuario y el valor deberá estar dentro del rango de 1 – 2.5m.
- El tiempo de retención "Rt", deberá ser ingresado por el usuario y deberá estar dentro de un rango de 6 – 30 días.

## 3.2. Desarrollo del Software

El desarrollo del software se lo realizo utilizando “Microsoft Visual Studio 2010”, que es un Entorno de Desarrollo Integrador (IDE por sus siglas en ingles), que permite crear aplicaciones, sitio y aplicaciones web utilizando una gran variedad de lenguajes programación.

El uso de este software de desarrollo se lo realizo por las extensas ventajas que ofrece al momento de desarrollar un proyecto informático; entre los beneficios más destacados esta la facilidad de programación, lo intuitivo que es crear un entorno y la ayuda directa que da el software a medida que se avanza en el desarrollo.

### 3.2.1. Desarrollo de la Programación Interna

Primero creamos el nuevo proyecto de “Aplicación de Windows Forms” y lo nombramos; una vez creado procedemos a crear una nueva carpeta dentro del proyecto a la que nombraremos como “clases” dentro de la cual ubicaremos los archivos de programación interna o base de cada uno de los procesos de la planta de tratamiento de aguas.

En la carpeta “clases” agregamos elementos de “C#” denominados igualmente clases cada elemento contiene la programación interna de un proceso de la planta de tratamiento, así tenemos 6 elementos de clases: Inhoff, fafa, desarenador, rejillas, parshall y el de la laguna anaerobia que al ser las formulas base iguales para los tres tipos de laguna nos sirve esa misma base para los tres diseños.

En estos elementos de programación interna o base denominados como “clases” vamos a definir primeramente las variables y posteriormente las ecuaciones o cálculos fundamentales de cada proceso, para ellos usamos los siguientes términos o palabras clave:

- **Float:** palabra clave de programación que se usa para almacenar o asignar valores de hasta 32 bits, por ejemplo si lo usamos de la siguiente manera “float x” esto significa que a “x” se le asignara un valor posteriormente; si lo usamos de la

siguiente manera “float x = 2 o float x = a + b”, significa que en x se almacenara o el valor asignado o al valor programado resultante de la suma de las variables.

- **Double:** al igual que float esta palabra clave se usa para asignar valores con la diferencia que este es capaz de asignar valores de hasta 64 bits.
- **Int:** esta palabra clave denota que el valor de una variable será entero es decir no puede contener decimales.
- **Public:** la palabra clave public es un modificador de acceso que permite tener un acceso libre a los datos, por ejemplo “public float x” esta expresión nos indica que los datos de la variable “x” no tienen restricción para ser leídos en cualquier parte del programa.
- **Private:** la palabra clave “private” es también un modificador de acceso pero esta restringe el acceso de los datos para que puedan ser leídos únicamente dentro del cuerpo de programación de la clase o de la estructura dentro de la cual se lo declara, por ejemplo “private float x” al designar de esta manera la variable x esta solo puede ser leída dentro del archivo de clase en el que se declaró esa expresión o dentro de la estructura definida en la que se lo declaro.
- **If:** es una sentencia de condición que generalmente se usa conjuntamente con “Else” para completar la sentencia de condición que funciona de la siguiente manera  

```
if (n > 20) { Console.WriteLine("Result1"); }  
else { Console.WriteLine("Result2"); },
```

Que se equivale a decir “si el valor de n es mayor a 20 se aplica el resultado 1, caso contrario se aplica el resultado 2”
- **Math:** proporciona constantes y métodos estáticos para operaciones trigonométricas, logarítmicas y otras funciones matemáticas comunes; por ejemplo “Math.Sqrt x” significa que el valor de x se somete a raíz cuadrada, también tenemos “Math.Sin, Math.Tan, Math.Cos” que usamos para someter una variable o valor a las operaciones trigonométricas.
- **Void:** esta palabra clave se utiliza para declarar que el método descrito a continuación no devuelve un valor y que se usa únicamente para determinar otras variables, por ejemplo si decimos “public void x = a+b” estamos declarando que el cálculo o serie de cálculos no debe expresar un valor sino que únicamente se usa

para determinar o asignar el valor de x y que a su vez este valor de x al estar definido bajo el parámetro de public puede ser utilizado para otros fines.

- **Llaves { }:** son usados para definir el inicio y el fin de un bloque de códigos o un bloque de datos.

Una vez definidos las diferentes palabras claves usadas procedemos a la definición de las variables que serán usadas tanto en cálculos como para expresar resultados de la siguiente manera:

**Ilustración 8:** Ejemplo de la definición de variables dentro de la programación

```
class Rejillas
{
    public float q;
    public float ancho;
    public float angulo;
    public float e;
    public float s;

    public float l;
    public float n;

    public Rejillas()
    {
```

*Autor: Cristian Suárez*

Y de la misma manera los cálculos internos:

**Ilustración 9:** Ejemplo de la programación para que el programa realice los cálculos

```
public void Calcular()
{
    float a = q / 0.6f;
    float h = a / ancho;
    l = h / (float)Math.Sin(ConvertToRadians(angulo));
    n = ((ancho*100) + e) / (s + e);
}
```

*Autor: Cristian Suárez*

En los que claramente podemos ver que se definen simplemente las formulas principales del proceso, y también se nombran otras variables que no fueron nombradas previamente ya que estas son usadas únicamente dentro del cálculo.

### **3.2.2. Desarrollo de la Interface de Cada Proceso**

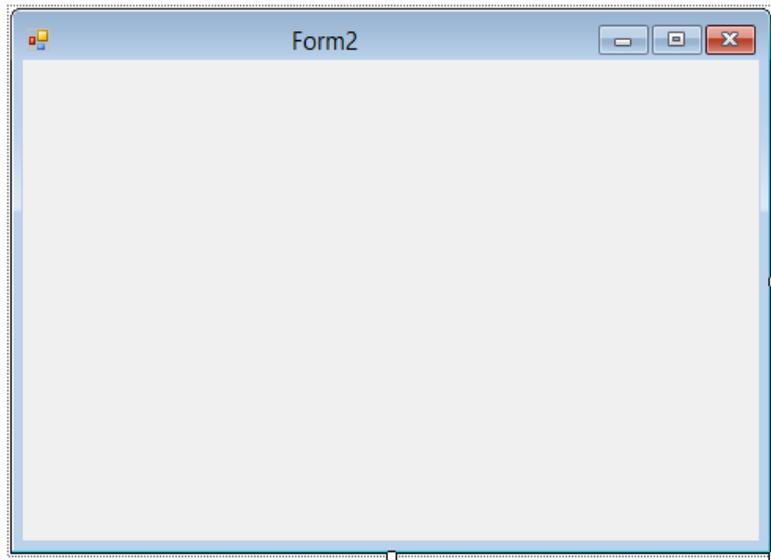
Para la interface vamos a agregar dentro del proyecto un nuevo elemento llamado “Windows Forms” agregaremos uno por cada proceso.

Este elemento de la programación constituye una ventana de Windows en blanco dentro de la cual podemos agregar todo lo que a la interface se refiere, así tenemos los siguientes elementos usados para el desarrollo del software:

- **Button:** herramienta que nos permite agregar botones, posteriormente a estos botones se les tiene que programar la función para la que están destinados.
- **Label:** sirve para presentar información ya sea como etiquetas informativas o para mostrar datos resultantes de la ejecución del programa.
- **TextBox:** permite crear cuadros de texto en los que el usuario puede ingresar información o datos necesarios para la ejecución del programa.
- **DataGridView:** esta herramienta permite crear tablas para mostrar datos organizados en filas y columnas.
- **PictureBox:** esta herramienta crea un cuadro dentro del cual podemos establecer imágenes.
- **RichTextBox:** permite mostrar o modificar el contenido dinámico, incluidos párrafos, imágenes, tablas, etc.
- **ToolTip:** permite implementar una pequeña ventana emergente que muestra una breve descripción de la función de un elemento cuando el usuario sitúa el puntero sobre el control.

Con todas las herramientas definidas procedemos a crear la interface:

## Ilustración 10: Ventana de WindowsForms en blanco



Autor: Cristian Suárez

Comenzamos con la ventana de WindowsForms en blanco y agregamos los elementos necesarios ya definidos:

## Ilustración 11: Ventana de WindowsForms con la interface ya realizada

Rejillas

Es una estructura formada por laminas o varillas en paralelo, con una disposición e inclinación acorde a las necesidades.

El proposito fundamental de las rejillas es el de retener materiales de gran tamaño, que podrían atascar o dañar, equipos o valvulas en etapas posteriores.

Se puede aplicar como etapa inicial en procesos cuyo fluido contenga solido de gran tamaño en el flujo.

Caudal  m<sup>3</sup>/s      Ancho  m           

Angulo  grados      Espesor (s)  cm     

Espacio (e)  cm

	Largo (L)	Numero de Barras
*		

L label8

C label9

label10

e

S label11

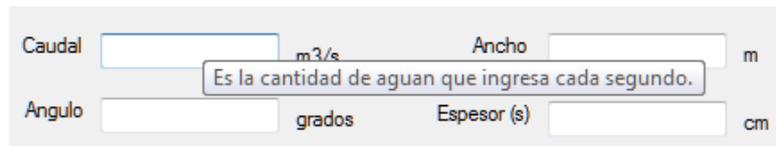
label11

label11

Autor: Cristian Suárez

Mediante la utilización de “Rich TextBox” se agregó una descripción del proceso; también se utiliza el elemento “ToolTip”, que es una utilidad que permite agregar ventanas de ayuda para que el usuario obtenga información específica al momento de ingresar los datos en cada cuadro con solo ubicar el puntero sobre el cuadro del que desea obtener información.

### Ilustración 12: ToolTip usado para proporcionar información



Autor: Cristian Suárez

Los “label” no definidos son usados para expresar los resultados del cálculo y los definidos muestran texto informativo como podemos ver a lado de cada cuadro de texto.

Los cuadros de texto cada uno deben ser renombrados dentro de sus propiedades para poder distinguirlos y así poder obtener los datos que ingresa el usuario de manera adecuada, por ejemplo el cuadro de texto del caudal será renombrado como txtCaudal y de esta manera en la programación del cálculo podremos leer el dato del caudal correctamente para los cálculos, lo mismo se hace con la tabla de datos, así tenemos:

### Ilustración 13: Ejemplo de la programación usada para leer los datos ingresados por el usuario

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    Rejillas r = new Rejillas();
    try
    {
        float q = float.Parse(txtCaudal.Text);
        float c = float.Parse(txtAncho.Text);
        float angulo = float.Parse(txtAngulo.Text);
        float s = float.Parse(txtEspesor.Text);
        float espe = float.Parse(txtEspacio.Text);

        r.ancho = c;
        r.q = q;
        r.angulo = angulo;
        r.s = s;
        r.e = espe;
    }
}
```

Autor: Cristian Suárez

Aquí se puede observar que al aplastar el botón “Calcular” el programa leerá primeramente los datos de los cuadros de texto, como se había dicho por ejemplo el cuadro de texto de caudal (txtCaudal), y manda los valores a las variables definidas dentro de la programación interna.

También observamos los botones “Guardar” e “Imprimir”; ambos funcionan de manera semejante haciendo una captura de pantalla del proceso que se está ejecutando en ese instante con los datos ingresados así como los resultados. El botón “Imprimir” manda esa captura a la impresora conectada para así facilitarnos una copia física de lo realizado; y el botón “Guardar” está vinculado a “saveFileDialog”, que es un cuadro de dialogo que nos ayuda en el proceso de guardado de la captura en el cual podemos seleccionar en directorio de guardado, así como el nombre del archivo a guardar.

También para facilitar el proceso de guardado, dentro de la programación de la interface se define el tipo de documentos que se van guardar, con ello evitamos problemas de guardado o lectura posteriores.

**Ilustración 14:** Línea de código usada para definir el formato en que se realizara el guardado

```
saveFileDialog1.Filter = "image files (*.jpg)|*.jpg|All files (*.*)|*.*";
```

Autor: Cristian Suárez

Además de esto se definen nuevas variables que observamos con una “r.” previo a la variable esto lo hacemos para mandar las respuestas a los diferentes cuadros y tablas.

**Ilustración 15:** Ejemplo de la programación realizada par amostrar los resultados de los cálculos

```
r.Calcular();

dgResultado.Rows[0].Cells[0].Value = r.l;
dgResultado.Rows[0].Cells[1].Value = r.n;

label8.Text = "" + r.l;
label9.Text = "" + r.ancho;
label11.Text = "" + r.s;
label10.Text = "" + r.e;
```

Autor: Cristian Suárez

Aquí tenemos la programación de los resultados en la que observamos efectivamente como se expresan los resultados y relacionando con el gráfico de la interface nos ayuda a comprender de mejor manera y que efectivamente los datos están saliendo en el lugar que corresponden.

Además en esta etapa se pueden definir todo tipo de limitantes y errores del programa en la programación del botón “Calcular”, antes de que pasen al cálculo interno y con ello a un gasto innecesario de recursos.

**Ilustración 16:** Programación usada para definir condiciones dentro de los cálculos y los mensajes de error

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    try
    {
        Anaerobia a = new Anaerobia();
        a.q = float.Parse(txtCaudal.Text);
        a.h = float.Parse(txtAltura.Text);
        a.angulo = float.Parse(txtAngulo.Text);
        a.ancho = float.Parse(txtAncho.Text);
        a.tr = float.Parse(txtTr.Text);

        if (tipo == Tipo.Anaerobia)
        {
            if (a.h >= 3 && a.h <= 6 && a.tr >= 10 && a.tr <= 50)
            {
                a.Calcular();
                dgResultados.Rows[0].Cells[0].Value = a.l;
                dgResultados.Rows[0].Cells[1].Value = a.a;
                dgResultados.Rows[0].Cells[2].Value = a.b;

                label11.Text = "" + a.l;
                label12.Text = "" + a.a;
                label13.Text = "" + a.b;
                label14.Text = "" + a.h;
                label15.Text = "" + a.hSeg;
            }
            else
            {
                MessageBox.Show("Error en el rango de los datos ingresados");
            }
        }
    }
}
```

Autor: Cristian Suárez

Aquí podemos observar una condición y una definición de error dentro de la programación del botón calcular, en la que nos indica claramente que si los datos ingresados de “h” que en este caso es la altura y de “tr” que es el tiempo de retención están dentro de los rangos definidos de 3 a 6 para la altura y de 10 a 50

para el tiempo de retención se procede a realizar los cálculos de la programación interna; caso contrario la programación “else” está definiendo que el programa mostrara un error con el mensaje “Error en el rango de los datos ingresados”, de esta manera definimos los rangos de trabajo del programa a la vez que evitamos cargar la memoria de la programación innecesariamente y le hacemos saber al usuario lo que está mal mediante el mensaje de error.

### 3.2.3. Desarrollo de la Interface General

Esta etapa al igual que la de la interface de los procesos se desarrolla en base al elemento “WindowsForms” y aquí simplemente se enumeran los procesos y se programa la ejecución inicial de cada uno.

**Ilustración 17:** Menú principal del software PTAR



**Autor:** Cristian Suárez

Aquí se agregó una breve descripción de lo que es una planta de tratamiento de aguas residuales; así como también una descripción de cada uno de los procesos enlistados para ello a cada botón se le programo una acción “MouseEnter”, con la cual al pasar el mouse sobre cualquier botón de proceso en la ventana de la

descripción aparecerá un texto explicando el proceso que estamos seleccionando según como se ha programado.

**Ilustración 18:** Ejemplo de la programación usada para agregar las descripciones

```
private void btnRejillas_MouseEnter(object sender, EventArgs e)
{
    lblDescripcion.Text = "Descripcion: \n \nEs una estructura formada por
}
```

Autor: Cristian Suárez

A cada botón se programa para que llame a ejecución a su correspondiente proceso programado.

**Ilustración 19:** Programación de botones para que ejecuten el proceso correspondiente

```
public partial class Menu : Form
{
    public Menu()
    {
        InitializeComponent();
    }

    private void btnParshall_Click(object sender, EventArgs e)
    {
        ParshallForm p = new ParshallForm();
        p.Text = "Parshall";
        p.ShowDialog();
    }

    private void btnRejillas_Click(object sender, EventArgs e)
    {
        RejillasForm r = new RejillasForm();
        r.Text = "Rejillas";
        r.ShowDialog();
    }
}
```

Autor: Cristian Suárez

Aquí podemos observar la programación de dos botones “Parshall” y “Rejillas” los mismos que hacen el llamado a su sub interface que se puede identificar por la terminación “form” y con ellos a su proceso previamente programado.

Además a cada botón de proceso se le agrego en sus propiedades una imagen que lo relacione con el proceso que calcula.

### **3.2.4. Desarrollo del Instalador**

El desarrollo del instalador constituye un proyecto básicamente automático en una serie de pasos los mismos que se realizan después de depurar el programa ya finalizado.

Para el instalador dentro de la solución o programa creamos un nuevo proyecto en mismo se agrega desde el menú nuevo proyecto/otros tipos de proyectos/instalación e implementación/instalación de visual studio y ahí se selecciona proyecto de instalación.

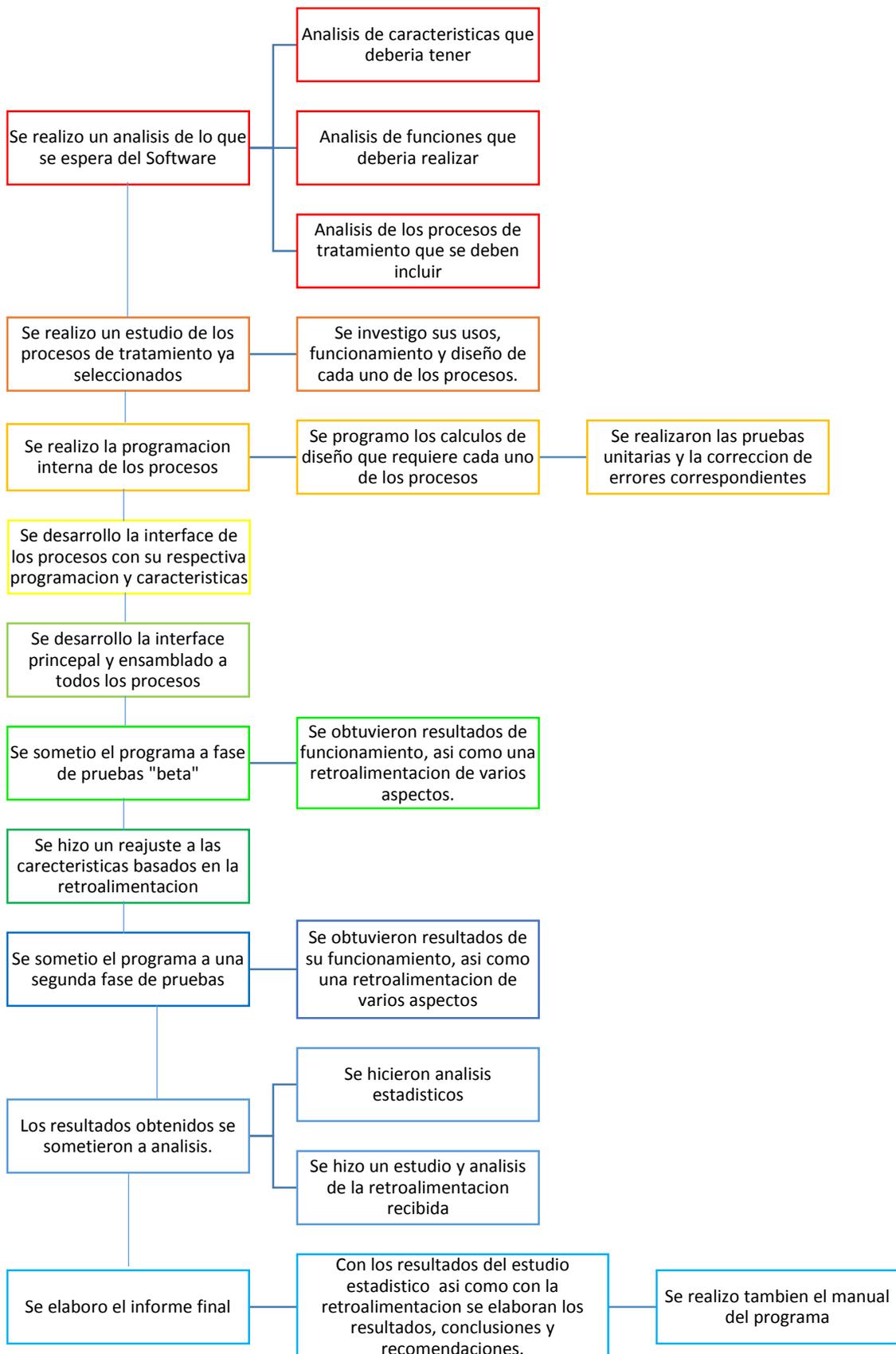
Nos genera una ventana con tres carpetas:

- **Carpeta de la aplicación:** en esta carpeta agregaremos los resultados del proyecto, es decir la aplicación ya finalizada y funcional.
- **Escritorio del Usuario:** en esta carpeta agregaremos el acceso directo a la aplicación que agregamos previamente en la “carpeta de la aplicación”, así mismo aquí podemos cambiar el nombre de acceso al nombre que queremos que se muestre de nuestro programa.
- **Menú Programas del Usuario:** en esta carpeta agregaremos otra carpeta con el nombre de nuestro programa y en esta carpeta de igual manera agregamos otro acceso directo de nuestro programa.

Una vez definidas las carpetas entramos a las propiedades del proyecto de instalación para poder definir y agregar los requisitos previos para el funcionamiento correcto de nuestro proyecto, en este caso los requisitos previos definidos para el funcionamiento son: Microsoft .Net Framework 4 y Windows Installer 4.5; además de definir los requisitos aquí definimos para que estos requisitos sean adjuntados al instalador principal de nuestro proyecto e instalados conjuntamente para que no haya inconvenientes, esto lo hacemos mediante la opción “Descargar los requisitos previos desde la misma ubicación que mi aplicación”.

De esta manera queda listo todo el paquete de instalación el último paso es generar el ejecutable, ello se logra desde el menú “generar” mediante la opción “Generar Setup”, con ello el programa nos generara los archivos necesarios para una instalación independiente.

### 3.3. Resultados



**Ilustración 20:** Diagrama de flujo del proceso de creación del Software

Autor: Cristian Suárez

Después del estudio y desarrollo de este software tenemos como resultado un programa con una interface sencilla y muy comprensible cuyos datos y resultados pueden ser interpretados correctamente inclusive sin conocimientos previos dado que los mismos se ven reflejados gráficamente para facilidad de comprensión y diseño.

Asignado bajo el nombre “PTAR”, Plantas de Tratamiento de Aguas Residuales, por sus siglas; y bajo la licencia MIT de software libre. (Anexo)

### **Validación del Software**

Se sometió a un grupo de 6 profesionales a una prueba y calificación del software, utilizando el Método Delphi, en el cual a los 6 profesionales se les explico el uso del software y sus características, además de explicarles en que consiste la comprobación que ellos van a realizar.

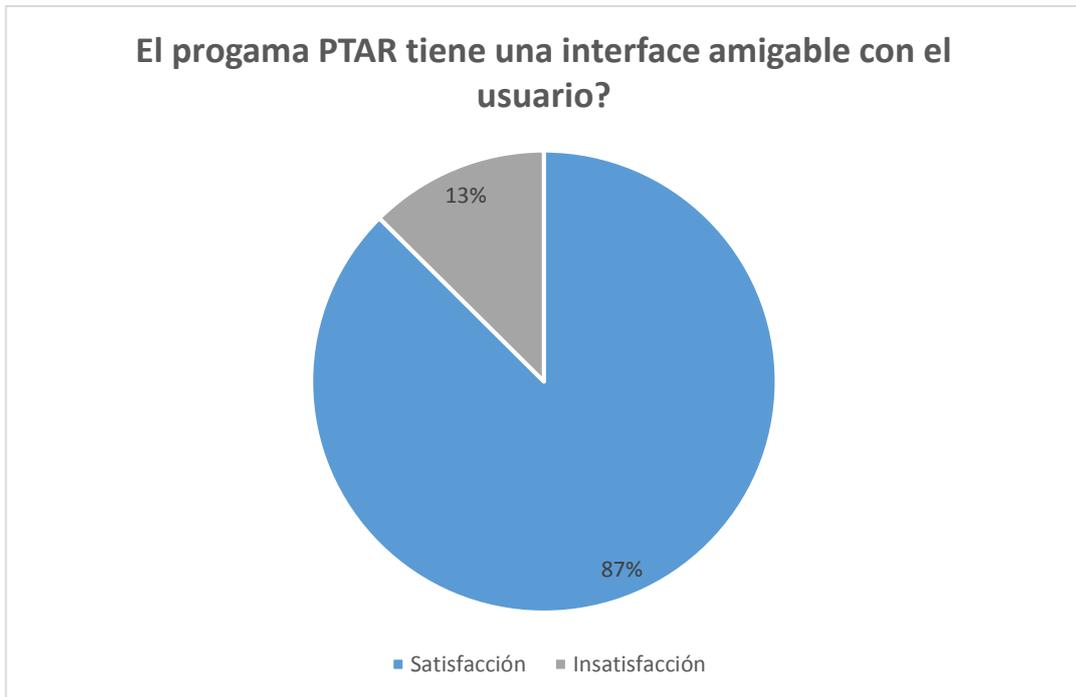
Los profesionales usaron el software y lo sometieron a las pruebas que ellos consideraron necesarias para revisar cada aspecto relevante del software y una vez concluidas sus pruebas se les facilito un cuestionario (anexo) para evaluar sus resultados y opinión.

**Ilustración 21:** Grafico que muestra el grado de satisfacción con la facilidad de uso del software



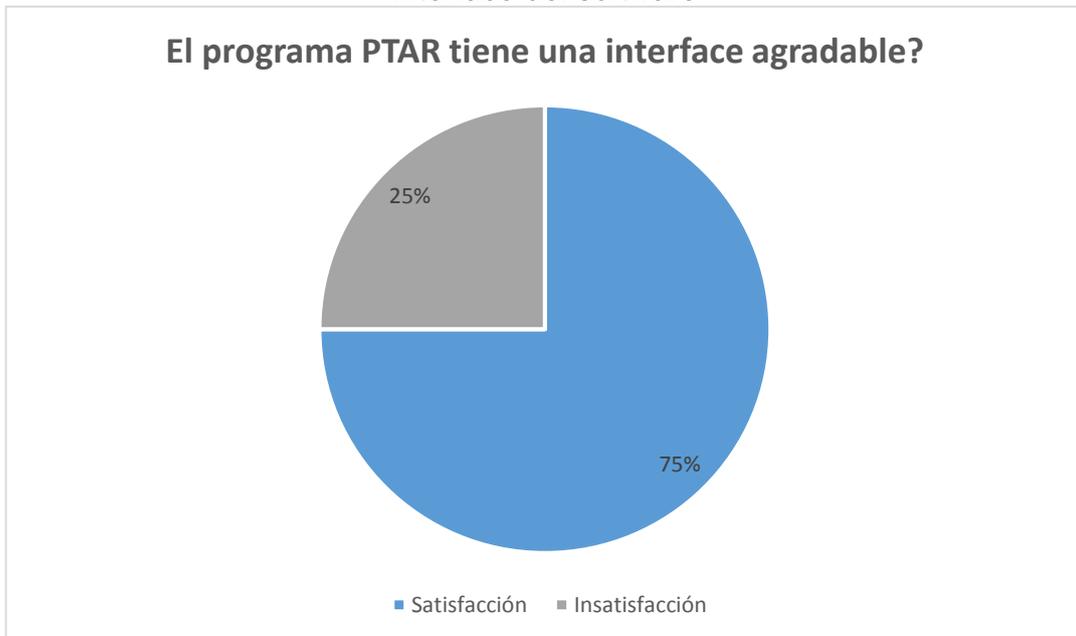
**Autor:** Cristian Suárez

**Ilustración 22:** Grafico que muestra el grado de satisfacción con el uso de la interface del software



Autor: Cristian Suárez

**Ilustración 23:** Grafico que muestra el grado de satisfacción con la estética de la interface del software



Autor: Cristian Suárez

**Ilustración 24:** Grafico que muestra el grado de satisfacción del uso del programa PTAR



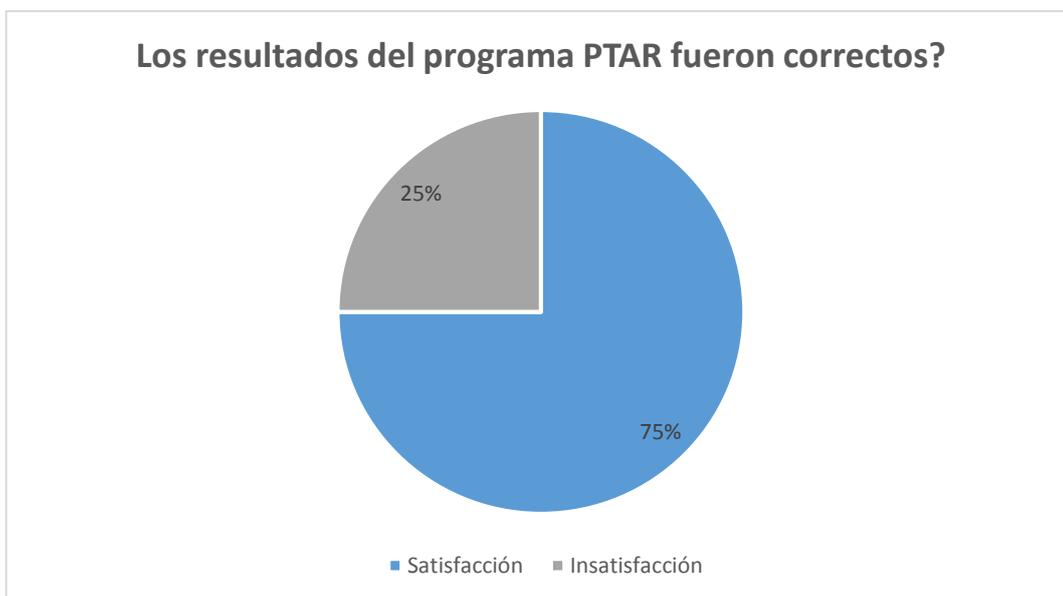
Autor: Cristian Suárez

**Ilustración 25:** Grafico que muestra la satisfacción con la manera en que el programa presenta los resultados



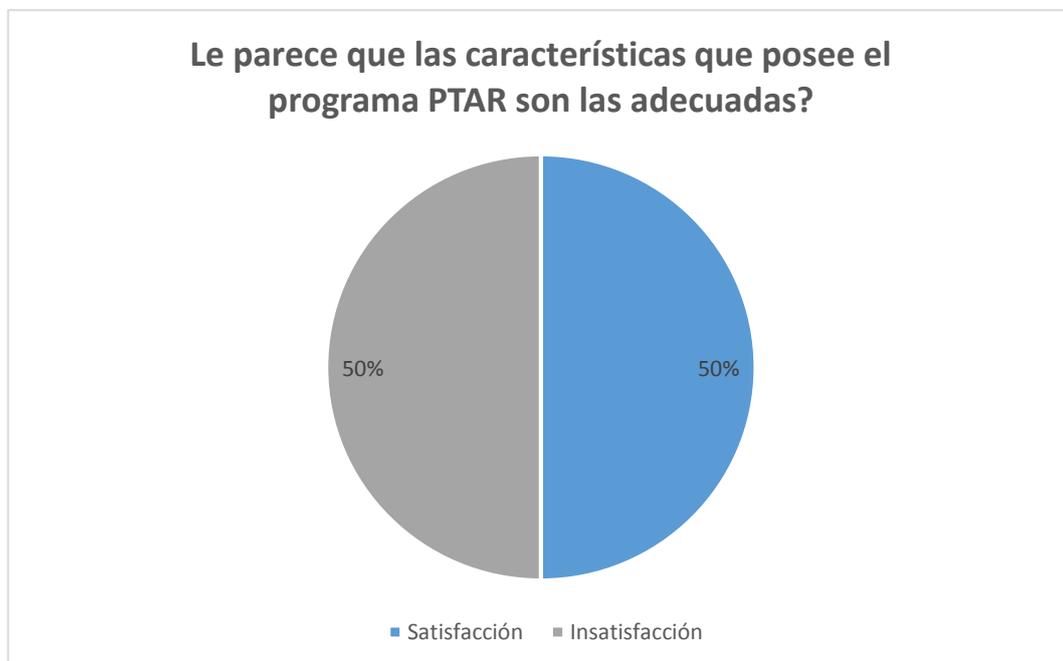
Autor: Cristian Suárez

**Ilustración 26:** Grafico del grado de satisfacción con la exactitud de los resultados



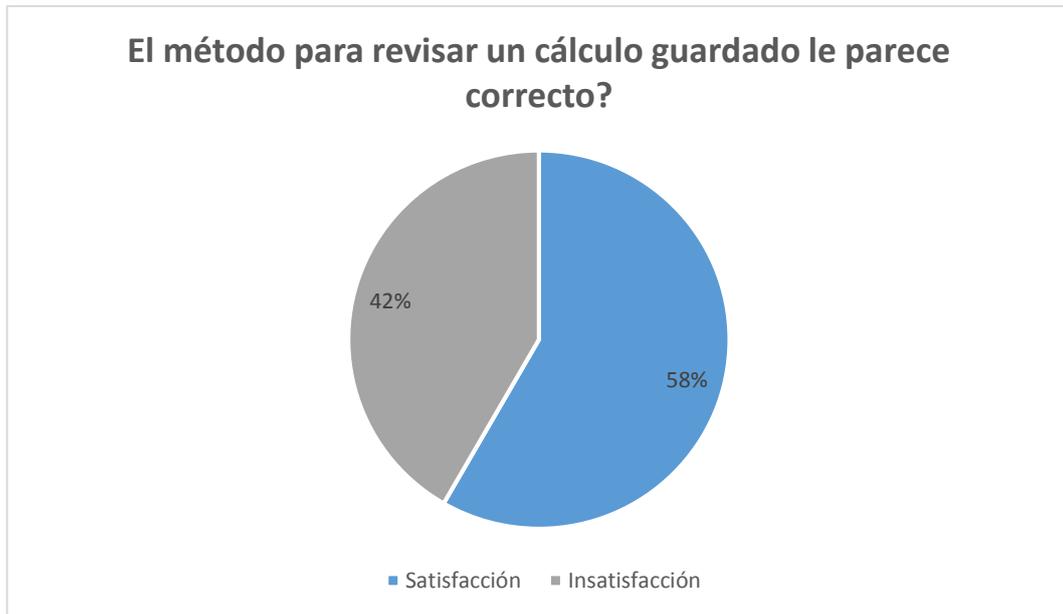
Autor: Cristian Suárez

**Ilustración 27:** Grafica del grado de satisfacción con las características del software



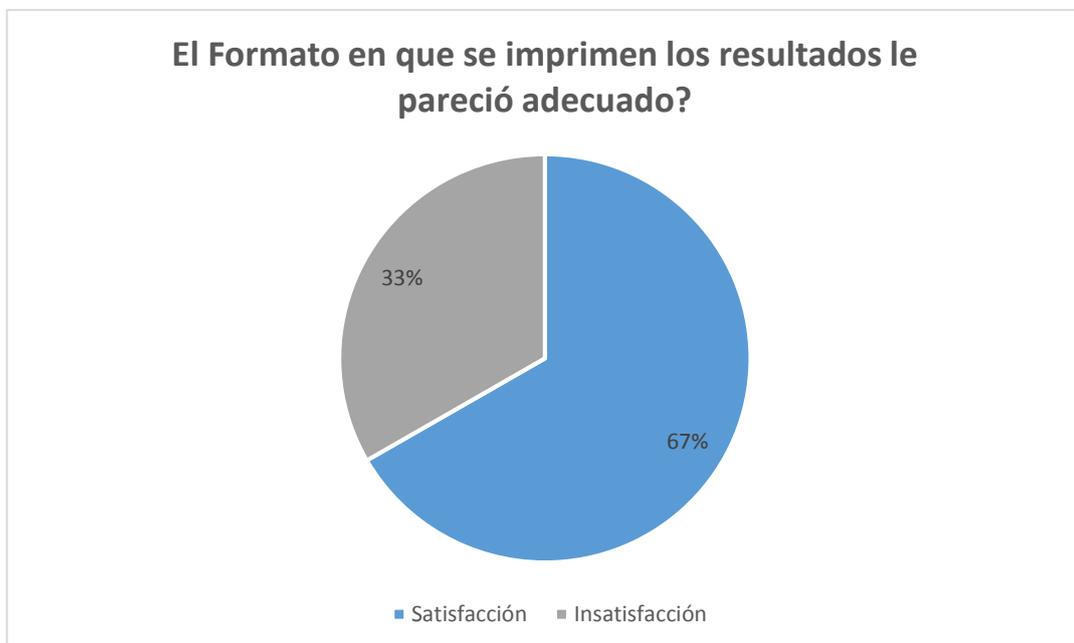
Autor: Cristian Suárez

**Ilustración 28:** Grafico del grado de satisfacción con el método de guardado de los cálculos



Autor: Cristian Suárez

**Ilustración 29:** Grafico de la satisfacción con el formato de impresión de los resultados del software



Autor: Cristian Suárez

El programa está destinado para uso didáctico en el cual se puede analizar cómo afecta la variación de los datos en el diseño de los procesos no cubre todos los procesos que pueden ser usados en plantas de tratamiento y además el programa solo está desarrollado para las etapas de diseño estructural con un nivel básico de cálculos de ingeniería, por lo cual cálculos adicionales, cálculos de comprobación y cálculos de desarrollo microbiano que pueden ser necesarias en ciertas etapas no están consideradas.

También se realizó un manual para facilitar su uso explicando adecuadamente sus aspectos y los resultados que obtendrá de su uso. (Anexo)

## CONCLUSIONES

- Se desarrolló el software con un estudio muy extenso de ambas áreas involucradas en el desarrollo tanto del área de informática principalmente programación como del área del tratamiento de aguas, esto especialmente porque en las dos áreas nos encontramos con diferentes y diversas metodologías de desarrollo, sin embargo con el debido estudio cada una de las etapas se diseñó adecuadamente.
- La gestión del diseño de la planta está distribuida en etapas individuales para una fácil manipulación de datos y resultados en una interface sencilla e intuitiva para facilitar su uso incluso a personas que no posea un conocimiento extenso de lo que es el diseño de una planta de tratamiento de aguas, el software especifica cada dato y medida, además de incluir gráficos que hacen que su comprensión sea más fácil.
- Se diseñó el software de tal manera que sin importar la persona o profesional que lo use, este obtenga resultados que sean de calidad y sobre todo facilitando el acceso a los mismos lo que significa que este se ajusta a cualquier tipo de necesidades que pueda tener cada profesional.

## RECOMENDACIONES

- Se recomienda un estudio más amplio de los métodos de diseño de plantas de tratamiento de aguas residuales, ya que durante el desarrollo y comprobación del proyecto nos encontramos con procesos con el mismo enfoque pero diferente metodología además de que en la mayoría de procesos se encuentran más cálculos para la comprobación para un adecuado funcionamiento.
- Es altamente recomendable que este tipo de proyectos se desarrolle por grupos de profesionales especializados y de amplio conocimiento para poder definir de mejor manera los procesos y metodologías más adecuados.
- Se podría implementar etapas de simulación tanto para la comprobación de funcionamiento, crecimiento microbiano, como de la planta en sí para un entendimiento total de lo que es cada proceso y la planta de tratamiento en sí.

## **BIBLIOGRAFÍA**

ÁLVAREZ, M. Qué Es La Programación Orientada A Objetos. 2001

<http://www.desarrolloweb.com/articulos/499.php>

2014-06-15

BAUTISTA, J. Programación Extrema (XP). 2014

[http://ingenieriadesoftware.mex.tl/52753\\_XP---Extreme-Programing.html](http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html)

2014-06-10

BUSTAMANTE, N. ¿QUÉ ES EL MÉTODO DELPHI?. 2012

<http://www.eoi.es/blogs/nataliasuarez-bustamante/2012/02/11/%C2%BFque-es-el-metodo-delphi/>

2014-10-30

CRITES, R., & TCHOBANOGLIOUS, G. (2000). Tratamientos de Aguas Residuales en Pequeñas Poblaciones., Bogotá - Colombia. McGraw Hill, pp. 33, 42 – 44, 46 – 48.

GIL, M. (2006). Depuración de aguas residuales: Modelización de procesos de lodos activos., Madrid – España. CSIC, pp. 129-150.

HAMMEKEN, A., & ROMERO., G. (2005). Análisis y Diseño de una Planta de Tratamiento de Aguas Residuales para el Municipio de San Andrés Cholupa. Universidad de Las Américas. Escuela de Ingeniería. Departamento de Ingeniería Civil., Puebla – México. pp. 34 – 37.

LEÓN SERRANO, Gonzalo. (1996). Ingeniería de sistemas de software. 4ta. ed., Madrid – España. Isdefe, pp. 24-30.

MANGA, T., MOLINARES, N., & ARRIETA, J. (2007). Tratamiento de aguas residuales mediante sistemas de lagunaje., Barranquilla – Colombia. Uninorte, pp. 100-140.

MATSUO, T., HANAOKI, K., TAKIZAWA, S., & SATOH, H. (2001). *Advances in Water and Wastewater Treatment Technology.*, Tokyo – Japon. Elsevier B.V, pp. 305-323.

METCALF., & EDDY. (1995). *Ingeniería de Aguas Residuales: Tratamiento, Vertido y Reutilización.* 3ra. ed., Madrid – España. McGraw-Hill, pp. 95 – 102

OROZCO, A. (2005). *Bioingeniería de Aguas Residuales.*, Bogotá – Colombia. Acodal, pp. 299-310.

PEPPER, I., GERBA, C., & GENTRY, T. (2014). *Environmental Microbiology.* 3ra. ed., San Diego – EE.UU. Academic Press, pp. 583-606.

PRESSMAN, Roger. (2002). *Ingeniería de software.* 5ta. ed., Madrid – España. McGraw – Hill, pp. 19-29, 224.

RANADE, V., & BHANDARI, V. (2014). *Industrial Wastewater Treatment, Recycling, And Reuse.*, Oxford - UK. Butterworth-Heinemann, pp. 60-80.

RIGOLA, M. (1989). *Tratamiento de aguas industriales: aguas de proceso y residuales.*, Barcelona – España. Marcombo, pp. 43-48.

ROJAS, R. (2002). *Sistemas de Tratamiento de Aguas Residuales.* CEPIS/OPS - OMS. Curso internacional: “Gestión integral de tratamiento de aguas residuales”. Lima – Perú. p. 19.

ROMERO, J. (2002). *Calidad del Agua.*, Bogotá - Colombia. Alfaomega, pp. 67 – 71, 74, 233 – 246, 706 – 707.

TUROFF, M., & LINSTONE, Harold, A. (1975). *Delphi Method: Techniques and Applications.*, Ann Arbor – EE. UU. Addison-Wesley Educational, pp. 3-9.

## ANEXOS

### ANEXO A: Tabla de criterios para diseñar lagunas

<b>Característica</b>	<b>Anaerobia</b>	<b>Aerobia</b>	<b>Facultativa</b>
Profundidad (m)	3 – 6	0.6 – 1.2	1 – 2.5
Tiempo de Retención (días)	10 – 50	4 – 12	6 – 30

**Fuente:** Metcalf y Eddy, 1996.

**ANEXO B:** Tabla de relación entre el caudal y el ancho de la Garganta del canal Parshall.

ANCHO DE GARGANTA  w, en mm	INTERVALO DE DESCARGA	
	MÍNIMO Q, (l/s)	MÁXIMO Q, (l/s)
76.2	0.77	32.1
152.4	1.50	111.1
228.6	2.50	251.0
304.8	3.32	457.0
457.2	4.80	695.0
609.6	12.10	937.0
914.4	17.60	1427.0
1219.2	35.80	1923.0
1524.0	44.10	2424.0
1828.8	74.10	2929.0
2133.6	85.80	3438.0
2438.4	97.20	3949.0

**Fuente:** Ackers, 1978

**ANEXO C:** Tabla de Geometría de los Canales Parshall, Ackers, 1978.

W	A	B	C	D	E	L	G	K	M	N	P	X	Y
76.2	467	457	178	259	457	152	305	25	-	57	-	25	38
152.4	621	610	394	397	610	305	610	76	305	114	902	51	76
228.6	879	864	381	575	762	305	457	76	305	114	1080	51	76
304.8	1372	1343	610	845	914	610	914	76	381	229	1492	51	76
457.2	1448	1419	762	1026	914	610	914	76	381	229	1676	51	76
609.6	1524	1495	914	1206	914	610	914	76	381	229	1854	51	76
914.4	1676	1645	1219	1572	914	610	914	76	381	229	2222	51	76
1219.2	1829	1794	1524	1937	914	610	914	76	457	229	2711	51	76
1524.0	1981	1943	1829	2302	914	610	914	76	457	229	3080	51	76
1828.8	2134	2092	2134	2667	914	610	914	76	457	229	3442	51	76
2133.6	2286	2242	2438	3032	914	610	914	76	457	229	3810	51	76
2438.4	2438	2391	2743	3397	914	610	914	76	457	229	4172	51	76

Medidas en milímetros.

La distancia “a”, medida desde donde se inicia la garganta hasta el punto de medición de  $h_1$  es igual a dos tercios de la distancia “A” que tiene de largo la pared convergente.

## **ANEXO D:** Licencia MIT usada en el programa "PTAR".

Copyright (c) 2014 Escuela Superior Politécnica de Chimborazo "ESPOCH"

Se concede permiso por la presente, de forma gratuita, a cualquier persona que obtenga una copia de este software y de los archivos de documentación asociados ("PTAR"), para utilizar el Software sin restricción, incluyendo sin limitación los derechos de usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar, y/o vender copias de este Software, y para permitir a las personas a las que se les proporcione el Software a hacer lo mismo, sujeto a las siguientes condiciones:

El aviso de copyright anterior y este aviso de permiso se incluirán en todas las copias o partes sustanciales del Software.

EL SOFTWARE SE PROPORCIONA "TAL CUAL", SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA, INCLUYENDO PERO NO LIMITADO A GARANTÍAS DE COMERCIALIZACIÓN, IDONEIDAD PARA UN PROPÓSITO PARTICULAR Y NO INFRACCIÓN. EN NINGÚN CASO LOS AUTORES O TITULARES DEL COPYRIGHT SERÁN RESPONSABLES DE NINGUNA RECLAMACIÓN, DAÑOS U OTRAS RESPONSABILIDADES, YA SEA EN UN LITIGIO, AGRAVIO O DE OTRO MODO, QUE SURJA DE O EN CONEXIÓN CON EL SOFTWARE O EL USO U OTRO TIPO DE ACCIONES EN EL SOFTWARE.

**ANEXO E:** Encuesta realizada en la comprobación con el Método Delphi

**CALIFICACIÓN DEL SOFTWARE “PTAR”**

**1. El programa PTAR le parece de fácil uso?**

Totalmente en desacuerdo.....  
En desacuerdo.....  
Ni de acuerdo ni en desacuerdo.....  
De acuerdo.....  
Totalmente de acuerdo.....

**2. El programa PTAR tiene una interface amigable con el usuario?**

Totalmente en desacuerdo.....  
En desacuerdo.....  
Ni de acuerdo ni en desacuerdo.....  
De acuerdo.....  
Totalmente de acuerdo.....

**3. El programa PTAR tiene una interface agradable?**

Totalmente en desacuerdo.....  
En desacuerdo.....  
Ni de acuerdo ni en desacuerdo.....  
De acuerdo.....  
Totalmente de acuerdo.....

**4. Comprendió el uso del programa PTAR?**

Totalmente en desacuerdo.....  
En desacuerdo.....  
Ni de acuerdo ni en desacuerdo.....  
De acuerdo.....  
Totalmente de acuerdo.....

**5. Comprendió la manera en la que el programa PTAR proporciona resultados?**

Totalmente en desacuerdo.....  
En desacuerdo.....  
Ni de acuerdo ni en desacuerdo.....  
De acuerdo.....  
Totalmente de acuerdo.....

**6. Los resultados del programa PTAR fueron correctos?**

- Totalmente en desacuerdo.....
- En desacuerdo.....
- Ni de acuerdo ni en desacuerdo.....
- De acuerdo.....
- Totalmente de acuerdo.....

**7. Le parece que las características que posee el programa PTAR son las adecuadas?**

- Totalmente en desacuerdo.....
- En desacuerdo.....
- Ni de acuerdo ni en desacuerdo.....
- De acuerdo.....
- Totalmente de acuerdo.....

**8. El método para revisar un cálculo guardado le parece correcto?**

- Totalmente en desacuerdo.....
- En desacuerdo.....
- Ni de acuerdo ni en desacuerdo.....
- De acuerdo.....
- Totalmente de acuerdo.....

**9. El Formato en que se imprimen los resultados le pareció adecuado?**

- Totalmente en desacuerdo.....
- En desacuerdo.....
- Ni de acuerdo ni en desacuerdo.....
- De acuerdo.....
- Totalmente de acuerdo.....

**10. Si tiene algún comentario, sugerencia u opinión sobre el software PTAR escríbalo a continuación.**

.....

.....

.....

.....

**ANEXO F: Manual de Usuario para el programa "PTAR"**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**



**SOFTWARE PARA EL DIMENSIONAMIENTO  
DE PLANTAS DE TRATAMIENTO DE AGUAS  
RESIDUALES**

# PTAR

Software para el dimensionamiento de plantas de tratamiento de aguas residuales.

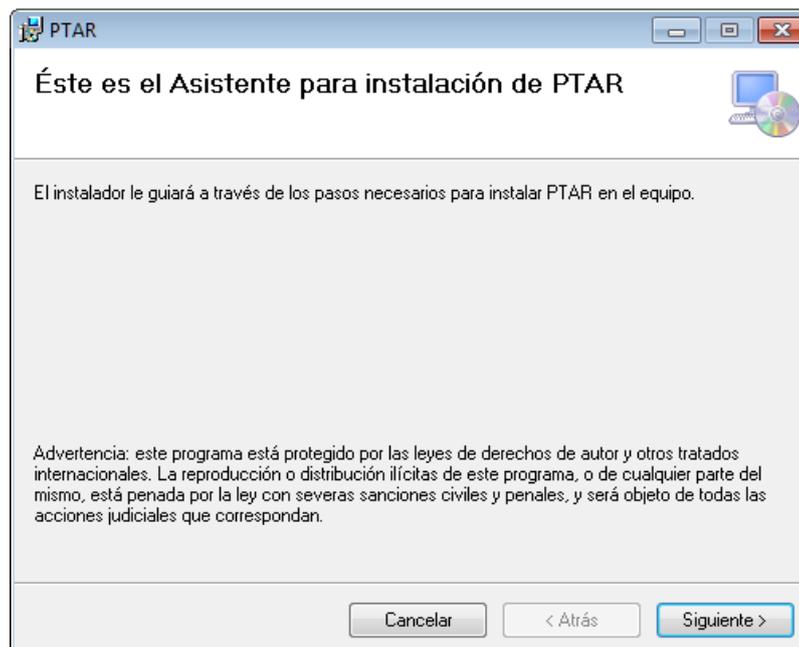
## Manual de usuario

### - Como instalar el software PTAR

1. Ingrese el medio que contenga el instalador (Cd, Memory Flash) al ordenador.
2. Explore el medio en busca de los archivos de instalación del Software PTAR.
3. Una vez ubicados damos doble clic en el archivo "setup.exe".

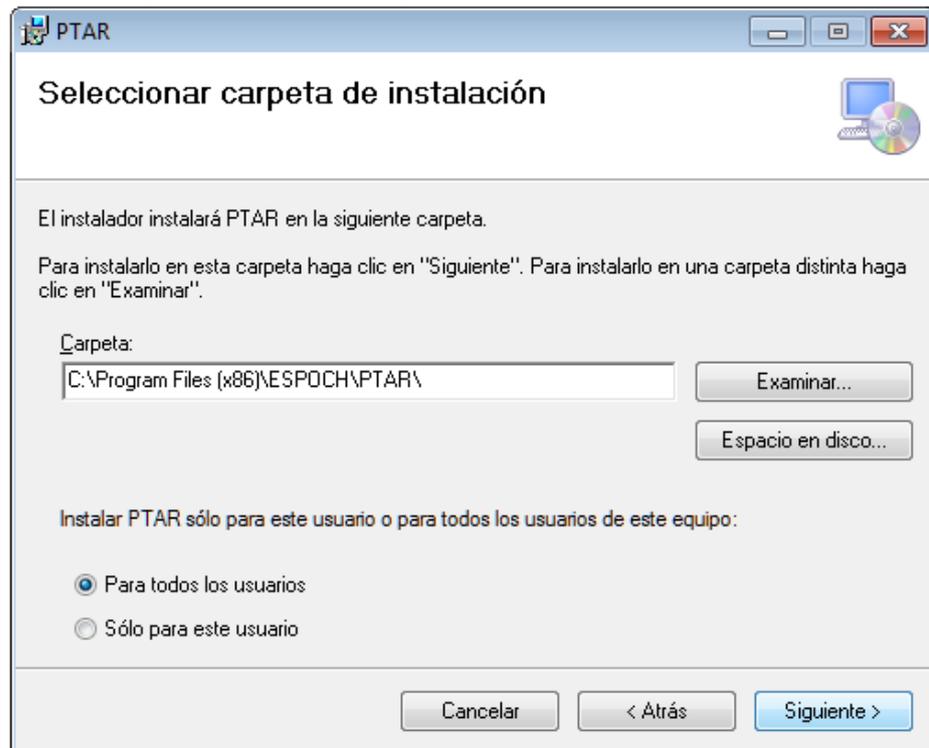
Nombre	Fecha de modifica...	Tipo	Tamaño
 DotNetFX40Client	29/10/2014 11:19	Carpeta de archivos	
 WindowsInstaller4_5	29/10/2014 11:19	Carpeta de archivos	
 PTAR.msi	29/10/2014 11:17	Paquete de Windo...	1.509 KB
 setup.exe	29/10/2014 11:17	Aplicación	472 KB

4. Aparecerá una nueva ventana del asistente de instalación, damos clic en el botón "Siguiente".



5. En la siguiente ventana seleccionaremos la carpeta en la que deseamos que se realice la instalación, la carpeta por defecto es "C:\Archivos de Programa\ESPOCH\PTAR\"; así también seleccionaremos los usuarios que tendrán acceso al programa en sus respectivos escritorios.

Una vez configurado esto damos siguiente.



6. En la siguiente ventana nos parecerá un mensaje de que el instalador está listo para instalarse, damos clic en siguiente y esperamos a que se instale el programa.

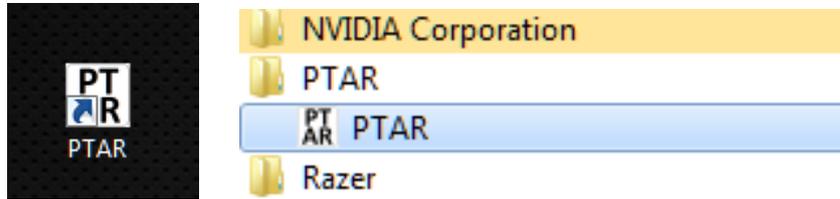
En versiones de Windows actuales (Vista, 7, 8 y 8.1) puede aparecer una ventalla de permiso para la instalación le damos "Si".

7. Una vez finalizada la instalación aparecerá una ventana que confirma que el programa se ha instalado correctamente, le damos en "cerrar".

Hecho esto tenemos listo el software para usarlo.

- **Como usar el Software PTAR**

1. Ya instalado podemos empezar a usar el programa para ello podemos ejecutarlo buscando su icono en el escritorio o en la lista de programas del menú inicio.



2. Una vez ingresado en el programa podremos observar el menú principal del Software.

En el encontraremos una breve descripción de las plantas de tratamiento, así como de cada uno de los procesos enlistados, para ello simplemente debemos mover el puntero del Mouse sobre los botones de los procesos.



3. Para acceder a cualquier proceso basta con dar clic sobre su icono y nos aparecerá una nueva ventana con el proceso seleccionado.

## - Canal Parshall

En el canal parshall podremos observar una descripción del proceso así como la interface de funcionamiento, dentro de la misma encontraremos cuadros para ingresar el caudal mínimo y el caudal máximo en (litros/segundo) necesarios para el dimensionamiento.

Parshall

Es una estructura de diseño calibrado, de forma abierta; tiene una sección convergente, una garganta y una sección divergente.

Los usos del canal parshall son:

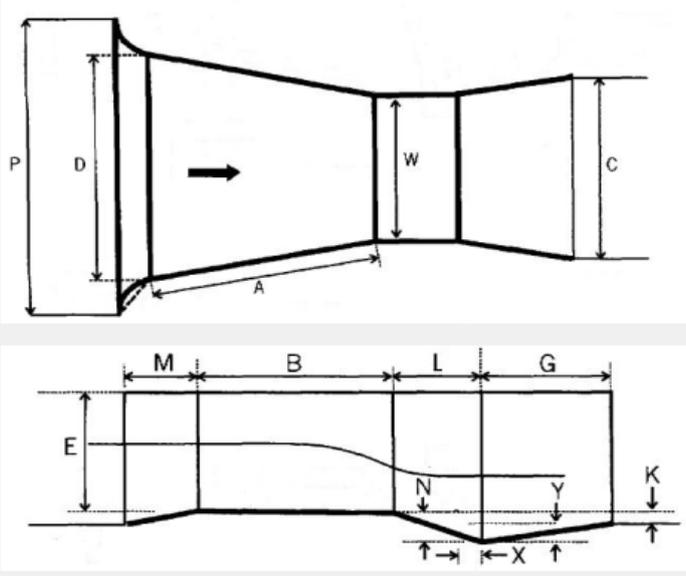
- Reducir la velocidad de entrada del flujo.
- Constituye una primera etapa de sedimentación.
- Sirve como estructura de medición

Y tiene la ventaja de operar en un rango amplio de flujos.

Pueden ser instaladas en todo tipo de procesos en los que se necesite disminuir la velocidad del fluido.

Ingrese Qmin:  l/s    Ingrese Qmax:  l/s           

	Qlnicial	Canales	W	A	B	C	D
*							



También encontramos 3 botones: el botón “aceptar” que procesa los datos ingresados para el dimensionamiento del canal, el botón de “imprimir” que hará una captura de pantalla de los cálculos y los imprimirá, el botón “guardar” que de igual manera hace una captura de pantalla de lo que hemos realizado y nos da la opción de guardarla dentro de nuestro equipo.

En la zona de resultados encontramos un cuadro deslizable en el cual observaremos los resultados obtenidos del cálculo; también tenemos una gráfica dentro de la cual podremos ver ubicados los valores resultantes en sus respectivos lugares para tener una idea del dimensionamiento.

## - Rejillas

En las rejillas podremos observar una descripción del proceso además la interface de funcionamiento, dentro de la misma encontraremos cuadros para ingresar el caudal en  $m^3/s$ , el ancho del canal en metros, el ángulo de inclinación de las rejillas, el espesor de las varillas y el espacio de separación entre varillas.

Rejillas

Es una estructura formada por laminas o varillas en paralelo, con una disposición e inclinación acorde a las necesidades.

El proposito fundamental de las rejillas es el de retener materiales de gran tamaño, que podrían atascar o dañar, equipos o valvulas en etapas posteriores.

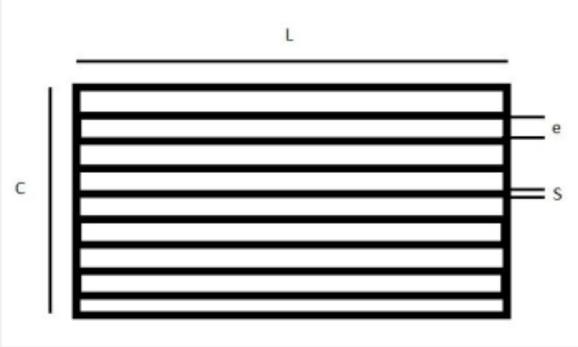
Se puede aplicar como etapa inicial en procesos cuyo fluido contenga solido de gran tamaño en el flujo.

Caudal   $m^3/s$  Ancho  m

Angulo  grados Espesor (s)  cm

Espacio (e)  cm

	Largo (L)	Numero de Barras
*		



Así mismo encontramos 3 botones: el botón “calcular” que procesa los datos ingresados para el dimensionamiento, el botón de “imprimir” que hará una captura de pantalla de los cálculos y los imprimirá, el botón “guardar” que de igual manera hace una captura de pantalla de lo que hemos realizado y nos da la opción de guardarla dentro de nuestro equipo.

En la zona de resultados encontramos un cuadro en el cual observaremos los resultados de longitud y numero fe barras obtenidos del cálculo; también tenemos una gráfica dentro de la cual podremos ver ubicados los valores resultantes en sus respectivos lugares para tener una idea del dimensionamiento.

## - Desarenador

En el dimensionamiento del desarenador encontramos una pequeña descripción del proceso además la interface de funcionamiento, en la que encontraremos cuadros para ingresar el caudal medio en  $m^3/s$ , el caudal máximo  $m^3/s$ , el caudal mínimo  $m^3/s$ , así como también el ancho del canal para un caudal normal en metros y el número de canales que se desea dimensionar.

**Desarenador**

Es una estructura, cuyo fin es la separación de la arena del agua, este procedimiento consiste en la reducción de la velocidad del agua por debajo de los límites de precipitación, pero sobre los límites de sedimentación de la materia orgánica.

Mediante su implementación se logra la precipitación de partículas con un diámetro superior a 2mm.

Su aplicación es recomendada para flujos con alto contenido de partículas sedimentables como arena o floculos.

T Normal  m      Q Max   $m^3/s$            

Q Med   $m^3/s$       Q Min   $m^3/s$      

Canales

	W	T Emer	T Normal	T Medio	T Min	H Emer	H Normal
*							

Diagrama de un desarenador que muestra la sección de control y la forma parabólica de la sedimentación. Se indican los caudales ( $Q$ ) y tiempos ( $T$ ) para emergente, normal, medio y mínimo, así como las alturas ( $H$ ) correspondientes. El ancho del canal es denotado como  $w$ .

También observamos 3 botones: el botón “calcular” que procesa los datos ingresados para el dimensionamiento, el botón de “imprimir” que hará una captura de pantalla de los cálculos y los imprimirá, el botón “guardar” que de igual manera hace una captura de pantalla de lo que hemos realizado y nos da la opción de guardarla dentro de nuestro equipo.

En la zona de resultados encontramos un cuadro deslizable en el cual observaremos los resultados obtenidos del cálculo; también tenemos una gráfica dentro de la cual podremos ver ubicados los valores resultantes en sus respectivos lugares para tener una idea del dimensionamiento.

- **FAFA**

En el dimensionamiento del Filtro “FAFA” encontramos una pequeña descripción del proceso además la interface de funcionamiento, en la que encontraremos cuadros para ingresar el caudal en  $m^3/día$ , el ancho del filtro que es un valor asumido o que propone el usuario en metros, así como también la altura útil del filtro en metros, el tiempo de retención en días y el número de filtros que se desea obtener.

The screenshot shows a software window titled "FAFA". It contains a descriptive text about the filter, followed by input fields for "Caudal" (m<sup>3</sup>/día), "Ancho" (m), "Altura Util" (m), "Tiempo Retencion" (días), and "Numero Filtros". There are three buttons: "Calcular", "Imprimir", and "Guardar". Below the buttons is a table with one row labeled "Largo" and a cell containing an asterisk (\*). At the bottom is a diagram of the filter structure, showing a cross-section with labels: "Entrada", "Medio de Soporte", "Falso fondo", "Salida", "Nivel libre del liquido", and "Hu". Dimensions are indicated as 0.3 m for the width (b) and the height of the support medium (Hu).

También observamos 3 botones: el botón “calcular” que procesa los datos ingresados para el dimensionamiento, el botón de “imprimir” que hará una captura de pantalla de los cálculos y los imprimirá, el botón “guardar” que de igual manera hace una captura de pantalla de lo que hemos realizado y nos da la opción de guardarla dentro de nuestro equipo.

En la zona de resultados encontramos un cuadro en el cual observaremos el valor del largo del tanque resultante del cálculo; también tenemos una gráfica dentro de la cual podremos ver ubicados los valores resultantes en sus respectivos lugares para tener una idea del dimensionamiento.

## - Tanque Imhoff

En el dimensionamiento del tanque "Imhoff" encontramos una descripción del proceso además seguido de la interface de funcionamiento, en la que encontraremos cuadros para ingresar el caudal en  $m^3/día$ , el ancho del canal principal en metros, además de la población que produce dicho caudal en número de habitantes, el tiempo de retención en días, la carga superficial en  $m/día$ , también se debe ingresar la dotación de lodos en  $m^3/habitante \times año$  y el ángulo de las paredes de sedimentación.

The screenshot shows a software window titled "Imhoff" with a descriptive text about the tank's function. Below the text are input fields for "Caudal" (m<sup>3</sup>/día), "Ancho" (m), "Poblacion" (habitantes), "Tiempo Retencion" (dias), "Carga Superficial" (m/día), "Dotacion Lodos" (m<sup>3</sup>/habitantes\*año), and "Angulo" (grados). There are three buttons: "Calcular", "Imprimir", and "Guardar". Below the input fields is a table with 8 columns labeled L, H, D, C, G, F, A, E and a row with an asterisk. At the bottom is a cross-sectional diagram of the tank with labels a through h and y through g.

	Largo (L)	H	D	C	G	F	A	E
*								

The diagram shows a cross-section of the tank with a central channel and two side chambers. The central channel has a sloped bottom with an angle  $\alpha$ . The side chambers have vertical walls. Labels a through h indicate different horizontal levels and widths, while y through g indicate vertical levels.

También observamos 3 botones: el botón "calcular" que procesa los datos ingresados para el dimensionamiento, el botón de "imprimir" que hará una captura de pantalla de los cálculos y los imprimirá, el botón "guardar" que de igual manera hace una captura de pantalla de lo que hemos realizado y nos da la opción de guardarla dentro de nuestro equipo.

En la zona de resultados encontramos un cuadro deslizable en el cual observaremos los valores resultantes del cálculo; también tenemos una gráfica dentro de la cual podremos ver ubicados los valores resultantes en sus respectivos lugares para tener una idea del dimensionamiento.

## - Lagunas Anaerobia, Facultativa y Aerobia

Por último se encuentra en el menú principal las diferentes clases de lagunas de tratamiento estas tres están construidas bajo los mismos cálculos y varían en los valores de profundidad y tiempo de retención.

Por lo cual si desarrollo dentro del software está desarrollado igualmente bajo la misma estructura pero son olvidar los límites que las diferencia a cada una.

En el dimensionamiento de las lagunas de tratamiento encontramos una pequeña descripción de lo que son estas lagunas seguido de la interface de funcionamiento, en la que encontraremos cuadros para ingresar el caudal en  $\text{m}^3/\text{día}$ , el ancho de las lagunas que es un valor que puede ser asumido o propuesto por el usuario en metros, además del tiempo de retención en días, la altura que es la profundidad de excavación de la laguna en metros y el ángulo de las paredes.

Las Lagunas son estanques cabados en el suelos; los mismos pueden contener bacterias: aerobias, anaerobias o facultativas dependiendo del tipo de laguna que sea.

Así mismo la profundidad como el tiempo de retencion de cada laguna varian segun su tipo.

Son utilizadas para tratar aguas con alto contenido de materia organica.

Caudal   $\text{m}^3/\text{dia}$  Ancho  m

Tiempo Retencion  dias Altura (h)  m

Angulo  grados

	Largo (L)	a	b
*			

Calcular Ayuda

Imprimir Guardar

Diagrama de la laguna:   
 - label12: ángulo de la pared izquierda   
 - a: ancho superior   
 - label11: longitud   
 - b: ancho inferior   
 - label13: ángulo de la pared derecha   
 - h seg: altura   
 - label14: altura de la excavación   
 - label15: altura de la laguna   
 - 3.5m: anchos de las paredes

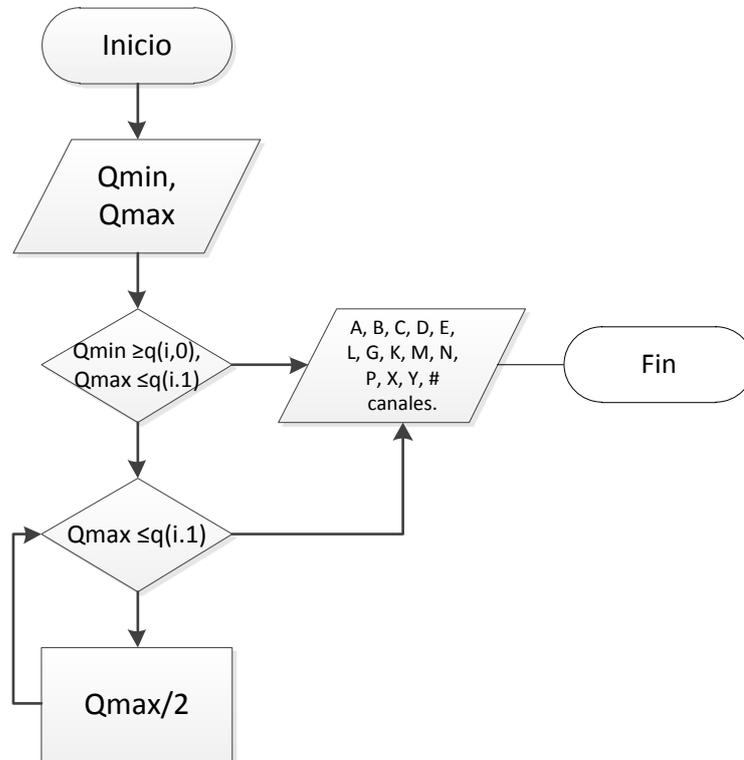
En el cálculo de las lagunas a diferencia de los demás procesos observamos 4 botones: el botón "calcular" que procesa los datos ingresados para el dimensionamiento, el botón de "imprimir" que hará una captura de pantalla de los cálculos y los imprimirá, el botón "guardar" que de igual manera hace una captura de pantalla de lo que hemos realizado y nos da la opción de guardarla dentro de nuestro equipo; y por último el botón de "Ayuda" el mismo que nos provee de los diferentes rangos permitidos de "Tiempo de retención" y "Altura" para cada uno de los tipos de laguna ya que si ingresamos mal estos valores el programa nos lanzara una pantalla de error.

En la zona de resultados encontramos un cuadro en el cual observaremos los valores resultantes del cálculo; también tenemos una gráfica dentro de la cual podremos ver ubicados los valores resultantes en sus respectivos lugares para tener una idea del dimensionamiento.

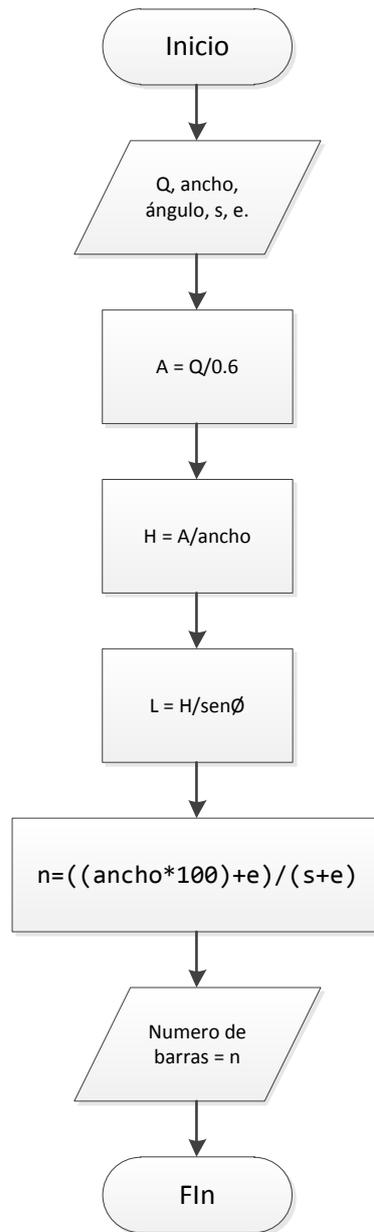
**ANEXO G:** Tabla de Selección de Procesos de Tratamiento

<b>PROCESO</b>	<b>POPULARIDAD</b>	<b>ECONOMÍA</b>	<b>CONSTRUCCIÓN</b>	<b>OPERACIÓN</b>	<b>MANTENIMIENTO</b>	<b>CÁLCULOS</b>
Canal Parshall	X	X	X	X	X	X
Rejillas	X	X	X	X	X	X
Filtro FAFA	X	X	X	X	X	X
Filtro de Turba	-	-	X	X	-	-
Reactor UASB	X	-	-	X	-	-
Tanque Imhoff	X	-	X	X	X	-
Lagunaje	X	-	-	X	-	X
Desarenador de Flujo Horizontal	X	X	X	X	X	X
Desarenador de Flujo Vertical	X	-	-	X	-	-
Desarenador de Flujo Inducido	-	-	-	X	-	-
Biodiscos	-	-	-	X	X	-

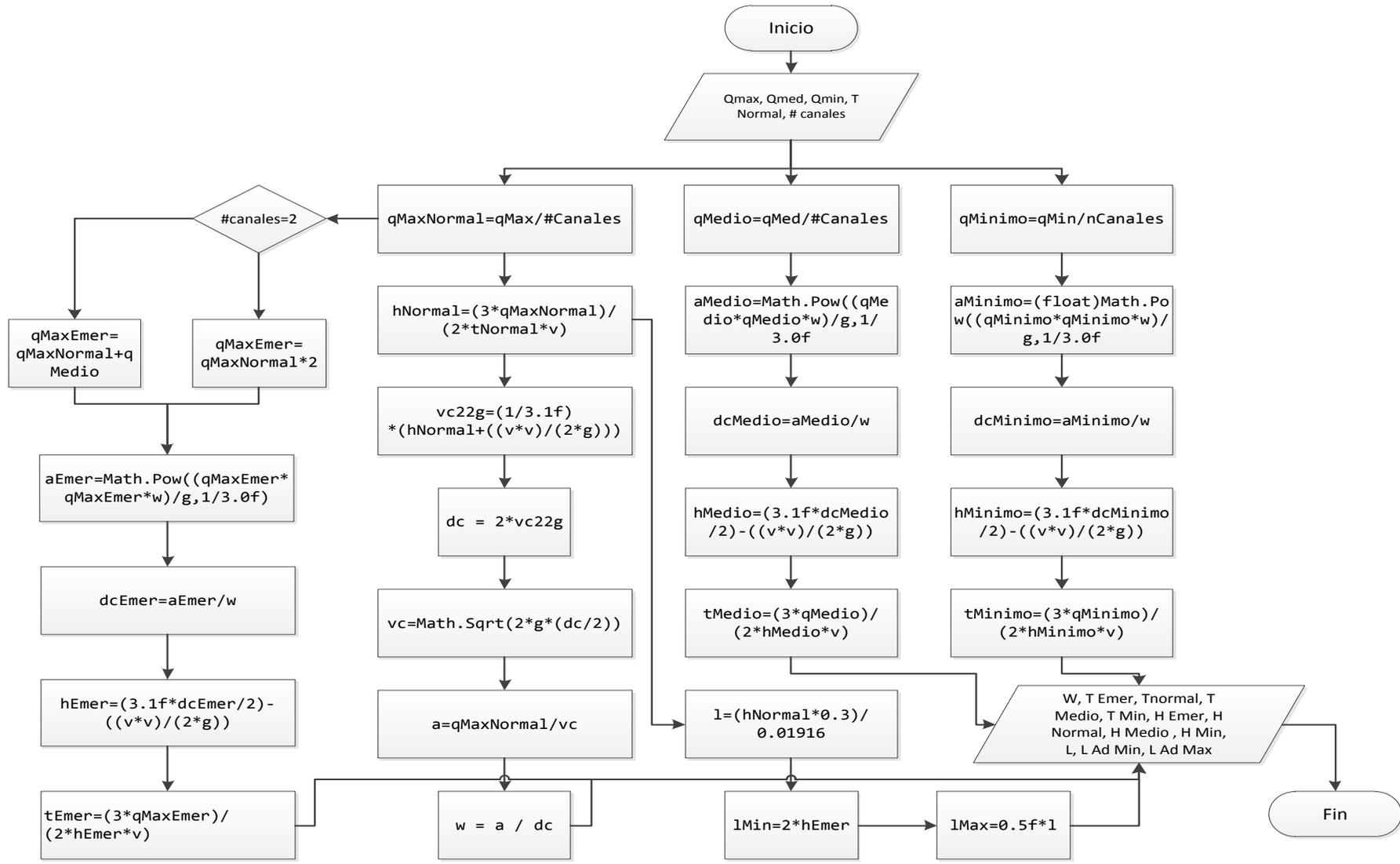
**ANEXO H:** Diagrama de flujo del proceso para el dimensionamiento del canal Parshall



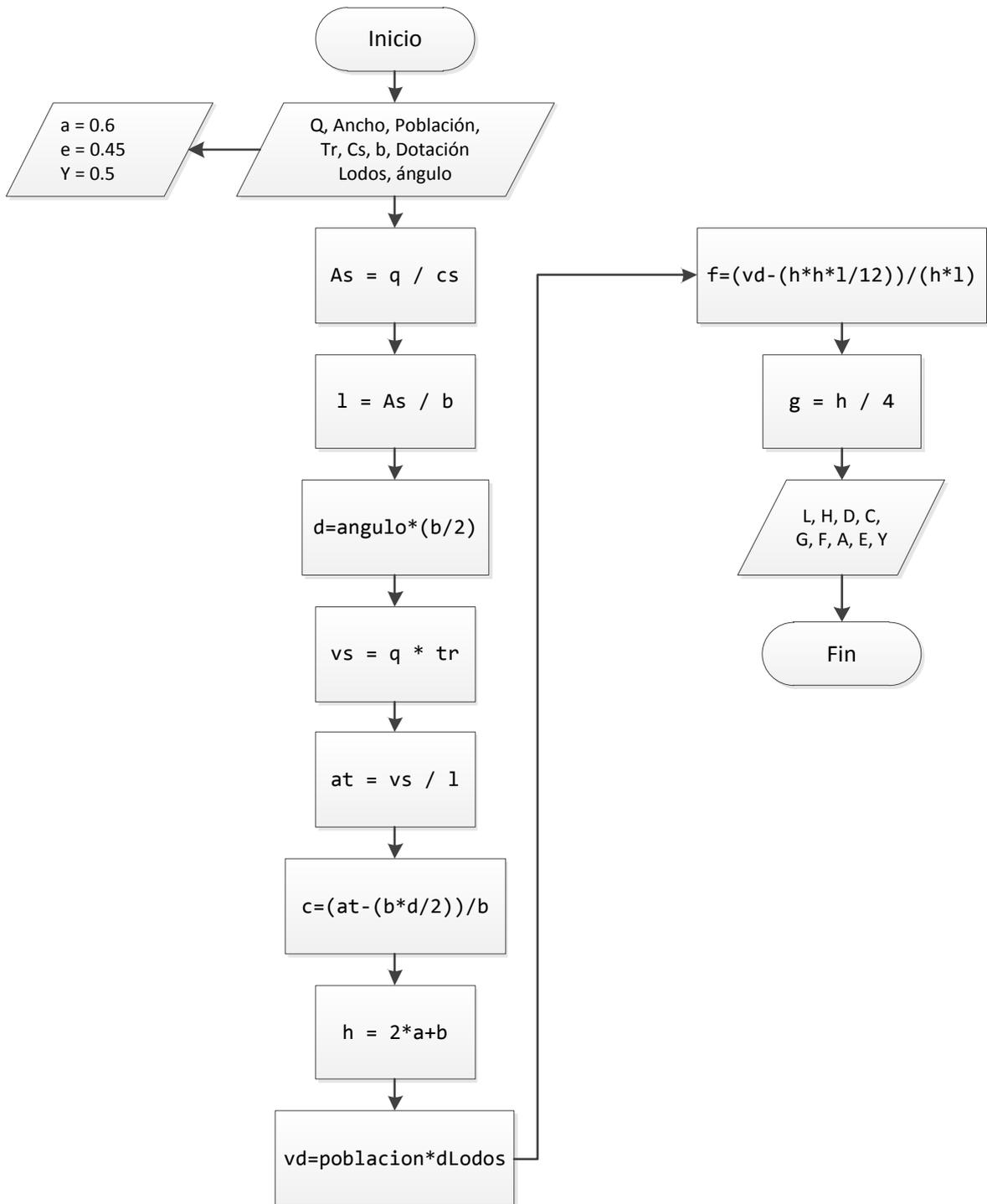
**ANEXO I:** Diagrama de flujo del proceso para el dimensionamiento de las Rejillas



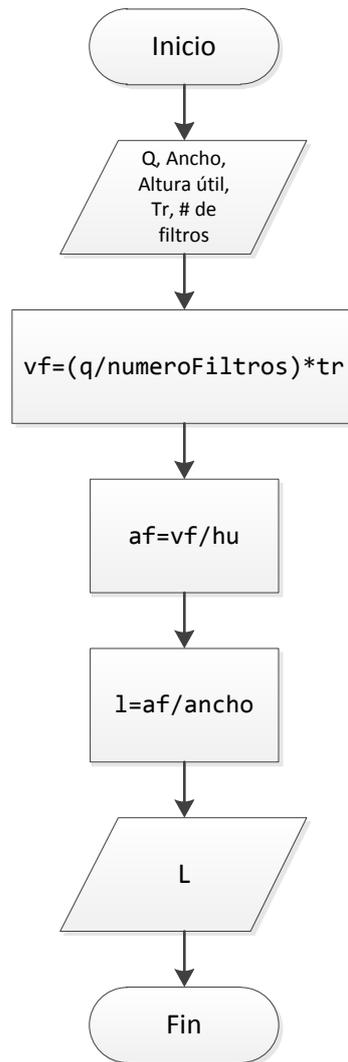
**ANEXO J: Diagrama de flujo del proceso para el dimensionamiento del Desarenador Horizontal**



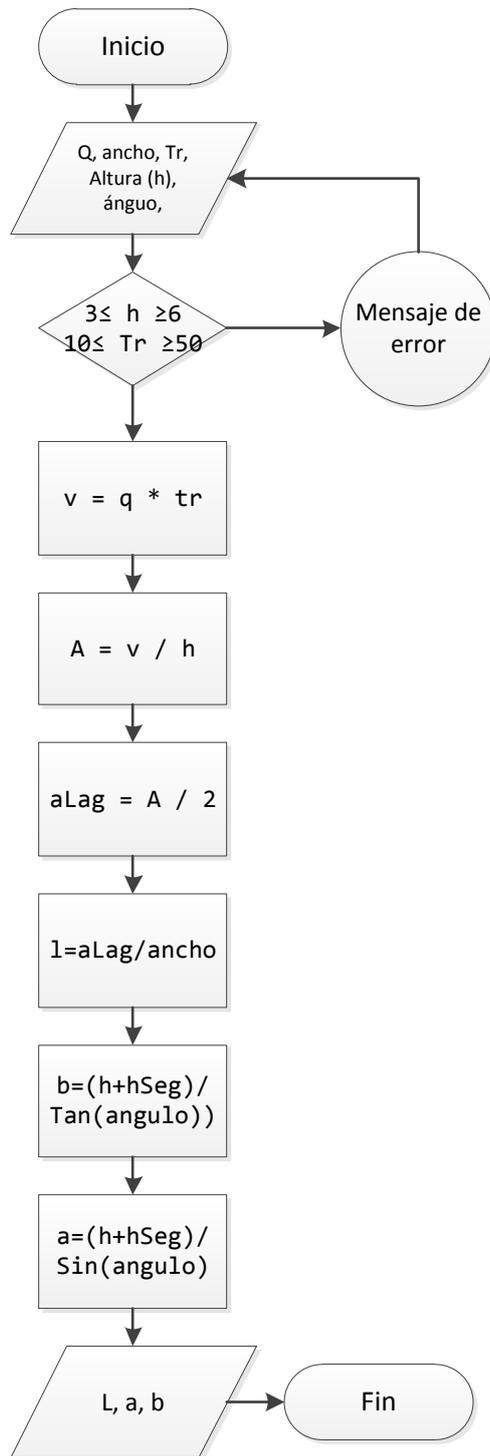
**ANEXO K:** Diagrama de flujo del proceso para el dimensionamiento del Tanque Imhoff



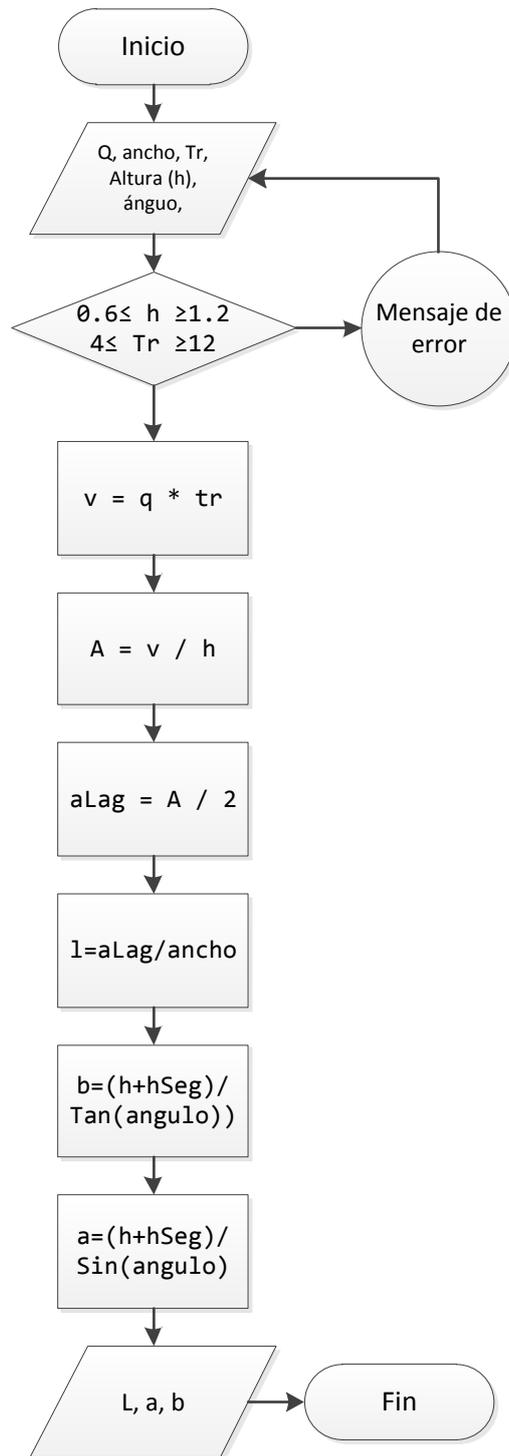
**ANEXO L:** Diagrama de flujo del proceso para el dimensionamiento del Filtro FAFA



**ANEXO M:** Diagrama de flujo del proceso para el dimensionamiento de la Laguna Anaerobia



**ANEXO N:** Diagrama de flujo del proceso para el dimensionamiento de la Laguna Aerobia



**ANEXO O:** Diagrama de flujo del proceso para el dimensionamiento de la Laguna Facultativa

